版权声明

A Mathematical Introduction to Logic, Second Edition by Herbert B. Enderton (ISBN: 0-12-238452-0).

ISBN: 981-259-674-7

译 者 序

数理逻辑作为基础数学的一个重要分支，在计算机科学中起着奠基作用，在模糊数学和人工智能等方面也都有着广泛的应用，有关它的教材和参考书有许多. 我们选择 Herbert B. Enderton 教授编写的这本数理逻辑教材是因为该书以可读性强而著称，在美国大学中采用率很高，是数理逻辑方面的经典教材之一. 同时，在第 2 版的修订中，作者增加了模型论和递归论的一些基础知识，其中有限模型、解析算法、有限计算和可判定性等内容都与计算机科学密切相关，这些内容是目前多数数理逻辑教材所没有的. 而这些知识，无论对于计算机专业还是基础数学专业的学生来说都是很重要的. 通过对这些内容的学习，学生能够加深对数理逻辑的了解，更多地接触数理逻辑在可计算性理论中的一些前沿应用.

本书除了内容上的改进之外，与传统教材相比，在编排方面也很有特点. 章节组织灵活，各章节内容相对独立. 读者会发现，不少章节的脚注中都有选学提示，可以根据个人需要选择适当的部分学习，而不会影响内容的连贯性，同时学有余力的读者也可进一步扩展知识面.

本书注重数理逻辑与其他数学分支的联系，引进了非标准分析、抽象代数和数论中的一些知识和例子，但这些内容的叙述并不晦涩难懂. 作者用清晰形象的语言阐述它们，大量独具匠心的解释使枯燥的定义、定理变得容易理解和接受. 既使熟悉数学的学生感受到数理逻辑与数学其他分支的紧密联系，又使不熟悉数学的学生对这些内容有直观的认识.

作者充分了解学生学习的难点和易犯的错误，增加了不少示例和解释. 用适当的示例结合一针见血的解释以及形象易懂的图形和图表，以不多的文字点出问题的关键所在.

基于上述特点，本教材适合于数学、哲学、计算机科学以及其他学科需要学习数理逻辑的本科生和研究生，同时对于以数理逻辑为工具从事研究工作的科研工作者也是一本很好的参考书. 我们相信，第 2 版中译本的出现能够让更多的师生在愉快的教与学中得到逻辑严谨美的享受.

由于数理逻辑主要研究形式语言，这种语言不同于我们平常使用的汉语、英语等自然语言. 在翻译的过程中，为了行文的连贯，我们将“英语中的……”译成“汉语中的……”. 一部分术语翻译成数理逻辑中常用的名词，除个别词之外，尽可能与数学、计算机科学中的名词一致.

本书由北京师范大学信息科学学院、哲学与社会学学院和经济学院的三位教师合作翻译而成. 由于译者水平有限，书中难免有疏漏和不妥之处，敬请广大师生、同行和专家批评指正!

前　　言

本书与第 1 版一样介绍了逻辑学中的基本概念和结果，主要包括证明、真值和可计算性. 与第 1 版相同，本书主要针对有一定数学背景知识并且对数理逻辑感兴趣的读者. 在这次修订中，我们做了许多“局部”改动，但关系到整本书的改动共有以下三处.

第一，我们力争使书中的内容容易为学生所接受. 在主要的改动中，我们不再把一些想法和知识认为是显然的，以免数学基础较薄弱的学生无法看懂.

第二，本书的结构更加灵活，以便于教师作为教材. 许多章节的开始部分都有脚注，为教师和读者的教学和学习提供可选择的方式.

第三，近年来，理论计算机科学对逻辑学产生了一定的影响. 在这次修订中，我们也突出了一些这样的影响：把可计算性问题作为重点内容之一，将有限模型的一些内容归入这一部分.

本书可以作为大学本科中、高年级学生学习数理逻辑课程的入门教材. 它涉及逻辑学中的一些重要概念和定理，并阐述了它们的重要性以及与数学其他一些分支的联系.

作为教材，本书适合用半个学期到一学年的课时来讲授. 半个学期一般能讲到一阶理论的模型 (2.6 节). 一个学期的话，富裕的时间，可以讲 3.0 节的不可判定性. 如果第二学期还有课程安排，那么就有时间讲授第 3 章的内容 (关于不可判定性).

本书适合没有学过逻辑学但有数学推理经验的读者阅读. 当然，我们希望读者还有一定程度的抽象能力. 在学习过程中，不可避免地要用到集合论的知识. 第 0 章对要用到的集合论知识做了简明概括. 读者可以先略过这一章，有需要时再回过来参考这些知识. 教师在授课时可以适当掌握集合论知识的使用，比如，基数可以完全不讲 (如果这样，一些定理也不用讲). 本书包括一些抽象代数中的例子，但它们只是例子，并不说明本质问题. 总体来说，第 3 章和第 4 章对读者能力的要求要比前两章高.

在 1.4 节中，对归纳和递归给出了更加深入的一些讨论，我们更愿意在课堂上只对这些问题给出非形式化的解释而在书中给出严格描述.

在每一节的末尾几乎都有习题. 如果习题的题号是黑体的，则说明这道习题的结果在前面的正文中讲评过. 通常较难的习题都标有星号.

我诚挚地感谢我所有的老师、同事和学生. 同时，我非常乐意接受任何来自读者的意见和建议. 本书的配套网址是 http://www.math.ucla.edu/~hbe/amil.

引　言

符号逻辑是演绎推理的数学模型. 应该说至少初期确实如此，但与其他数学分支一样，它的发展已经大大超出了最初的环境. 符号逻辑是一种模型，很大程度上就像现代概率论是可能性和不确定性的一种模型一样.

那么，这些模型是怎样建立起来的呢? 我们可以用建立现实生活中的具体对象的模型来解释，比如为一架飞机建模，那么我们选出建模的原对象要在模型中体现的一些特征，如飞机的形状，而忽略其他的特征，如飞机的大小等. 接下来，我们可以构建一个模型，这个模型在某些方面 (本质上的) 和原对象很相像，而在其他一些方面 (不相关的) 与原对象不同. 所建立的模型是否符合我们原来的要求，在很大程度上取决于特征选取.

逻辑要比飞机更加抽象. 现实生活中的对象都是某种“逻辑正确”的推理. 例如,

所有的人都是要死的.
苏格拉底是人.
所以苏格拉底是要死的.

第 3 句 (结论) 是从前两句 (假设) 中推出的，推理的正确性并不依赖于苏格拉底的特殊身份，而是取决于命题的形式，与“死”这个经验事实无关. 实际上,“死”是什么意思在这里并不重要，重要的是“所有的”这个词的含义.

⊚ 是 ★，只要它是 ♦.
一个东西是 ♦，并且它是 ⊚.
那么它是 ★.

尽管我们不知道 ★、⊚ 是什么，但我们同样知道第 3 个命题可以从前两个命题中推出.

逻辑正确的推理要比上面介绍的例子有趣得多. 实际上，公理化的数学就完全是由这样的推理组成的. 数学家实际给出的推理，被反映在我们的模型中.

这些推理的逻辑正确性源于它们的形式，而与它们的内容无关. 这种论断是模糊的，而正是这种模糊性促使我们转而研究推理的数学模型. 我们的主要目标是，在模型中，给出这一论断的准确的描述. 我们最关心的与模型有关的问题有:

(1) 什么叫作一个命题能够从其他命题“逻辑推出”?

(2) 如果一个命题的确能够从其他命题逻辑推出，应该采取什么样的方法来证明这个事实?

(3) 在公理系统中 (比如自然数的公理系统中)，可以证明的命题和在自然数中正确的命题是否相同?

(4) 逻辑和可计算性之间有什么联系?

实际上，我们将涉及两种模型. 第一种是命题逻辑，它虽然很简单却不适用于一些有趣的推理. 它的局限性使得它只能表达现实推理的一些梗概. 第二种模型是一阶逻辑，它适用于数学中遇到的推理. 当一个数学家断定某个特殊的命题可以从集合论的公理中推出时，他是指这个推理可以转换到我们的模型中.

本书内容的选择更侧重于与数学联系紧密的内容，而不包括多值逻辑、模态逻辑和直觉逻辑等. 这几种逻辑表现了现实推理的另一些不同的性质.

到现在为止，我们对将要研究的模型，比如一阶逻辑，并没有谈论太多. 作为简单的提示，下面我们简要地给出一些例子，看一看这种形式语言的表达能力. 第一个例子是集合论中的外延公理，“如果第一类对象中的元素和第二类对象中的元素相同，那么这两类对象相同.” 我们可以把它写成一阶语言

$$\forall x \forall y(\forall z(z \in x \leftrightarrow z \in y) \to x = y).$$

第二个例子对于学过微积分的同学来说是很熟悉的. “对于所有正数 ε，存在一个正数 δ 使得对于任何与 a 的差小于 δ 的数 x，$f(x)$ 与 b 的差小于 ε.” 可以写成

$$\forall \varepsilon(\varepsilon > 0 \to \exists \delta(\delta > 0 \land \forall x(\mathrm{d}xa < \delta \to \mathrm{d}fxb < \varepsilon))).$$

这两个例子可以大体表明本书要研究的内容. 我们还需声明：有些内容我们并不打算讨论，以免产生误解. 本书并不打算教读者如何去思考，“逻辑”这个词有时指思考训练，但在这里并不是这个意思. 读者已经知道如何思考. 我们的书给出的是一些值得思考的有趣概念.

目　录

第0章

集合基础

本书假定读者熟悉集合论的基础知识. 即便如此, 在这里我们还是要对即将用到的集合论的基础知识作一个简要的论述, 至少要说明一些记号的用法. 建议读者不要一开始就仔细阅读这部分内容, 而只需要在今后的章节中, 遇到不熟悉的集合论知识时参考一下就行了. 作者向大家推荐自己所著的《集合论基础》[1] 作为集合论方面的参考书 (参看本书最后的参考文献).

首先介绍一些对术语的说明. 贯穿全书, 我们使用了数学中标准的缩写. "■" 用于表示一个证明的结束; "如果……, 那么……" 类型的命题有时候会缩写为 "…… $\Rightarrow$ ……"; 逆蕴涵则对应使用 "…… $\Leftarrow$ ……" ("蕴涵" 一词在数学中有特定含义); "当且仅当" 缩写为 "iff" (这已经成为数学语言的组成部分) 或符号 "$\Leftrightarrow$"; "因此" 缩写为 "$\therefore$".

"$x \neq y$" 是 "$x = y$" 的否定, "$x \notin y$" 是 "$x \in y$" 的否定, 类似这样的符号记法可以推广到其他情形. 比如, 1.2 节中定义了 "$\Sigma \vDash \tau$", 那么就用 "$\Sigma \nvDash \tau$" 表示其否定.

集合(set) 是指一些对象的全体, 这些对象称为集合的元素或成员. 通常, "$t \in A$" 表示 t 是 A 的元素, "$t \notin A$" 表示 t 不是 A 的元素. "$x = y$" 表示 x 和 y 是同一个元素, 也就是说符号 x 和 y 是同一个元素的不同名字. 如果 $A = B$, 那么对于任一元素 t, 都有 "$t \in A$ iff $t \in B$", 这是因为 A 和 B 是相同的. 反之, 我们考虑其外延: 如果 A 和 B 是两个集合, 且对于任意的元素 t 都有

$$t \in A \quad \text{iff} \quad t \in B,$$

那么就有 $A = B$. 这反映了集合的基本思想: 集合是由其元素所确定的.

在集合中加入新元素是非常有用的操作. 对于集合 A, $A; t$ 表示一个新的集合, 其元素包括 (i)A 的元素和 (ii) 元素 t(可能是新的), 这里的 t 可能属于也可能不属于 A. 使用后面定义的符号, 这个过程可以表示为

$$A; t = A \cup \{t\}$$

并且

$$t \in A \quad \text{iff} \quad A; t = A.$$

空集 $\varnothing$ 是一个特殊的集合, 它不包含任何元素. 除此以外的其他集合都称为 非空的. 对于任意的对象 x, 存在单元素集合 $\{x\}$, 其唯一的元素就是 x. 更一般地, 对于任意有限个元素 $x_1, x_2, \cdots, x_n$, 都存在集合 $\{x_1, x_2, \cdots, x_n\}$, 此集合中的元素恰好就是这几个元素. 注意 $\{x, y\} = \{y, x\}$, 这是因为两个集合恰好含有相同的元素, 只不过是用不同的形式表示同一个集合. 如果一定要考虑元素的顺序, 那么可以使用有序对来表示 (稍后讨论).

1. 该书英文影印版已由人民邮电出版社出版, ISBN 7-115-14550-4/TP.5269.

集合的这种记法可以推广到某些简单的具有无限个元素的情形. 比如, $\{0,1,2,\cdots\}$ 是自然数集 $\mathbb{N}$, $\{\cdots,-2,-1,0,1,2,\cdots\}$ 是整数集 $\mathbb{Z}$.

记法 "$\{x|_x_\}$" 用于表示一个集合, 其元素 x 满足某种性质 $_x_$, 比如, $\{\langle m,n\rangle| m<n, m,n\in\mathbb{N}\}$ 表示第 1 个数比第 2 个数小的自然数的所有有序对的集合. $\{x\in A|_x_\}$ 就表示 A 中所有满足性质 $_x_$ 的元素的集合.

如果集合 A 的所有元素都是集合 B 的元素, 我们称 A 是 B 的子集, 记作 "$A\subseteq B$". 注意, 任何集合都是其自身的子集, 空集 $\varnothing$ 是每个集合的子集. ("$\varnothing\subseteq A$" 是 "毋庸置疑的", 因为要证明 $\varnothing$ 中的每个元素都属于 A 不需要做任何事, 或者从另外一个角度考虑, 仅当 A 的某个元素不属于 B 时 $A\subseteq B$ 才不成立, 如果 $A=\varnothing$, 这是不可能的.) 我们可以从集合 A 得到一个新的集合—— A 的 **幂集**(power set)$\mathcal{P}A$, 其元素是 A 的所有子集. 这样,

2

$$\mathcal{P}A=\{x|x\subseteq A\}.$$

例如,

$$\mathcal{P}\varnothing=\{\varnothing\},$$
$$\mathcal{P}\{\varnothing\}=\{\varnothing,\{\varnothing\}\}.$$

A 与 B 的 **并集**$A\cup B$ 是属于 A 或属于 B 的元素的集合. 比如, $A;t=A\cup\{t\}$. 类似地, A 与 B 的 **交集**$A\cap B$ 是所有 A 与 B 共有的元素的集合. 集合 A 与 B**不相交**(disjoint), 当且仅当二者的交集为空集 (即二者没有公共元素). 一组集合称为是 **两两不相交的**(pairwise disjoint), 当且仅当其中任意两个集合都不相交.

更一般地, 考虑集合 A, 其中的元素也是集合. 并集 $\cup A$ 可以通过将 A 中所有元素放入一个集合来获得:

$$\bigcup A=\{x|\ x\text{属于 } A \text{ 的某个元素}\}.$$

类似地, 对于非空集合 A,

$$\bigcap A=\{x|\ x\text{属于 } A \text{ 的所有元素}\}.$$

比如, 如果 $A=\{\{0,1,5\},\{1,6\},\{1,5\}\}$, 那么

$$\bigcup A=\{0,1,5,6\},$$
$$\bigcap A=\{1\}.$$

下面是另外的两个例子:

$$A\cup B=\bigcup\{A,B\},$$
$$\bigcup\mathcal{P}A=A.$$

如果对每个自然数 n, 都有一个集合 A_n 与之对应, 那么这些集合的并集 $\bigcup\{A_n|n\in\mathbb{N}\}$ 通常记作 "$\bigcup_{n\in\mathbb{N}}A_n$" 或者 "$\bigcup_n A_n$".

元素 x 与 y 的有序对 $\langle x,y\rangle$ 定义如下:

$$\langle x,y\rangle=\langle u,v\rangle \text{ iff } x=u\text{且 } y=v.$$

所有具有上述性质的定义都可以作为有序对的定义, 其中, 一个标准的定义是

$$\langle x,y\rangle=\{\{x\},\{x,y\}\}.$$

3

有序三元组可以定义为

$$\langle x, y, z\rangle = \langle\langle x, y\rangle, z\rangle.$$

更一般地，对于 $n > 1$ 可以如下递归地定义 n 元组：

$$\langle x_1, \cdots, x_{n+1}\rangle = \langle\langle x_1, \cdots, x_n\rangle, x_{n+1}\rangle.$$

为方便起见，对 $n = 1$ 的情形，我们定义 $\langle x\rangle = x$；这样上式对于 n=1 也是成立的．我们称 S 是 A 中元素的 *有限序列*(finite sequence)(或 *有限串*(string)) 当且仅当对某个正整数 n，$S = \langle x_1, \cdots, x_n\rangle$，其中每个 $x_i \in A$．(有限序列常常被定义为某个特定的有限函数，但是这里的定义更方便我们今后的使用．)

有限序列 $S = \langle x_1, \cdots, x_n\rangle$ 的 *子段*(segment) 是指一个有限序列 $\langle x_k, x_{k+1}, \cdots, x_{m-1}, \cdots, x_m\rangle, 1 \leqslant k \leqslant m \leqslant n$．这个子段是初始段当且仅当 k=1，一个子段是 *真子段*(proper) 当且仅当该子段与 S 不同．

如果 $\langle x_1, \cdots, x_n\rangle = \langle y_1, \cdots, y_n\rangle$，那么容易证明 $x_i = y_i, 1 \leqslant i \leqslant n$．(对 n 使用归纳法进行证明，证明的基本思想是用有序对的概念．) 但是，如果 $\langle x_1, \cdots, x_m\rangle = \langle y_1, \cdots, y_n\rangle$，那么通常不一定会得到 $m = n$．因为每个有序三元组也是一个有序对．不过，我们断言，只有当某个 x_i 本身是 y_j 的一个有限序列或者某个 y_j 是 x_i 的有限序列时，m 和 n 才不相等．或者：

引理 0A 如果 $\langle x_1, \cdots, x_m\rangle = \langle y_1, \cdots, y_m, \cdots, y_{m+k}\rangle$，那么 $x_1 = \langle y_1, \cdots, y_{k+1}\rangle$．

证明 对 m 使用归纳法．如果 m=1，那么结论是显而易见的．

对于归纳步骤，设 $\langle x_1, \cdots, x_m, x_{m+1}\rangle = \langle y_1, \cdots, y_{m+k}, y_{m+1+k}\rangle$，那么有序对的第一部分应该是相等的，即 $\langle x_1, \cdots, x_m\rangle = \langle y_1, \cdots, y_{m+k}\rangle$．使用归纳假设即可证明引理成立． ■

例如，设 A 是一个集合，它的任何一个元素都不是由其他元素组成的有限序列．如果 $\langle x_1, \cdots, x_m\rangle = \langle y_1, \cdots, y_n\rangle$ 且每个元素 x_i 和 y_j 都在 A 中，那么由上述引理可得 $m = n$，于是，也有 $x_i = y_i$．

我们可以构造集合 A 与 B 的 *笛卡儿积*(Cartesian product) $A \times B$，它是所有序对 $\langle x, y\rangle$ 的集合，其中 $x \in A$，$y \in B$．A^n 表示 A 中元素构成的所有的 n 元组组成的集合．比如，$A^3 = (A \times A) \times A$．

关系(relation)R 是有序对的集合．例如，数字 0~3 上的大小序关系是有序对的集合：

$$\{\langle 0,1\rangle, \langle 0,2\rangle, \langle 0,3\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,3\rangle\}.$$

R 的 *定义域*(domain) 记作 dom R，它是指所有满足 $\langle x, y\rangle \in R$ 的元素 x 的集合，其中 y 是任意的；R 的 *值域*(range) 记作 ran R，它是指所有满足 $\langle x, y\rangle \in R$ 的元素 y 的集合，其中 x 是任意的．dom R 与 ran R 的并称为 R 的 *域*(field)，记作 fld R．

4

A 上的 n 元关系是 A^n 的子集．若 $n > 1$，它就是一个关系；不过当 $n = 1$ 时，A 上的一元关系只是 A 的一个子集．A 上的一个特殊的二元关系是恒等关系 $\{\langle x, x\rangle | \, x \in A\}$．对于 A 上的 n 元关系 R 和 A 的一个子集 B，R 对 B 的限制是指交集 $R \bigcap B^n$．例如，上例中的关系是 $\mathbb{N}$ 上的序关系对集合 $B = \{0, 1, 2, 3\}$ 的限制．

函数 一个具有 *单值* 性质的关系 F：对于定义域 dom F 中的每一个 x，都有唯一的一个 y 满足 $\langle x, y\rangle \in F$．通常，这个唯一的 y 称为 F 在 x 上的值 $F(x)$．(这个记法是欧拉首先采

用的，遗憾的是，他没有使用 $(x)F$ 来表示函数的值，$(x)F$ 的记法对于复合函数的表示是非常有用的：要计算 复合(composition) 函数 $f \circ g$ 在 x 的值 $f(g(x))$，首先要计算 g 在 x 的值，再计算 f 在 $g(x)$ 的值.）

我们称 F 将 A 映射(map) 到 B 中，记作

$$F : A \to B,$$

意味着 F 是一个函数，$\text{dom } F = A$, $\text{ran } F \subseteq B$. 若 $\text{ran } F = B$, 则称 F 将 A 映射到 B上. F 是一个 一对一映射(one-to-one) 当且仅当对于 $\text{ran } F$ 中的每个 y, 都存在唯一的一个 x 使得 $\langle x, y\rangle \in F$. 如果 $\langle x, y\rangle$ 在定义域 $\text{dom } F$ 中，那么记 $F(x, y) = F(\langle x, y\rangle)$. 这个记法可以推广到 n 元的情形：$F(x_1, \cdots, x_n) = F(\langle x_1, \cdots, x_n\rangle)$.

A 上的 n元运算(n-ary operation) 是一个将 A^n 映射到 A 中的函数. 比如，加法是一个 $\mathbb{N}$ 上的二元运算，而后继运算 S(这里 $S(n) = n + 1$) 是 $\mathbb{N}$ 上的一元运算. 如果 f 是 A 上的 n 元运算，那么 f 在 A 的子集 B 上的限定是一个函数 g，其定义域为 B^n，且 g 在 B^n 中的每个点上的取值都与 f 在这些点上的取值相等. 这样，

$$g = f \cap (B^n \times A).$$

这个函数 g 是 n 元运算当且仅当 B 在 f 的作用下是 封闭的，即 $f(b_1, \cdots, b_n) \in B$，只要其中的每个 b_i 都在 B 中. 在这种情况下，$g = f \cap B^{n+1}$，这与关系的限制的定义是一致的. 例如，$\mathbb{N}$ 上的加法运算 (包含类似于 $\langle\langle 3, 2\rangle, 5\rangle$ 这样的三元组) 是 $\mathbb{R}$ 上的加法运算在 $\mathbb{N}$ 上的限制，而 $\mathbb{R}$ 中所包含的三元组要多得多.

A 上的一种特殊的一元运算是 恒等(identity) 函数 Id,

$$Id(x) = x, \quad x \in A.$$

因此，$Id = \{\langle x, x\rangle | x \in A\}$.

对于关系 R，我们定义：

- R 在 A 上是 自反的，当且仅当对 A 中每个 x 都有 $\langle x, x\rangle \in R$.
- R 是 对称的，当且仅当如果 $\langle x, y\rangle \in R$，则 $\langle y, x\rangle \in R$.
- R 是 传递的，当且仅当如果 $\langle x, y\rangle \in R$ 并且 $\langle y, z\rangle \in R$(若很巧)，则 $\langle x, z\rangle \in R$.

5

R 在 A 上满足 三分律(trichotomy)，当且仅当对 A 中任意的 x 和 y，如下 3 种可能中有且仅有一种成立：$\langle x, y\rangle \in R$, $x = y$, 或者 $\langle y, x\rangle \in R$.

R 是 A 上的 等价关系 当且仅当 R 是 A 上一个自反的、对称的和传递的二元关系.

R 是 A 上一个 序关系 当且仅当 R 是传递的且在 A 上满足三分律.

对于 A 上的等价关系 R 和 $x \in A$，我们定义 x 的 等价类 $[x]$ 为 $\{y | \langle x, y\rangle \in R\}$. 等价类对 A 进行了 划分，即每个等价类都是 A 的子集并且 A 的每个元素都恰好只属于一个等价类. 对 A 中的元素 x 和 y，

$$[x] = [y] \text{ iff } \langle x, y\rangle \in R.$$

自然数集 $\mathbb{N}$ 是集合 $\{0, 1, 2, 3, \cdots\}$. (自然数也可以用集合论的方法定义，见 3.7 节.) 集合 A 是 有限的 当且仅当存在一个函数 f 将集合 A 一对一映射到 $\{0, 1, \cdots, n-1\}$ 上. (可以将 f 看作是对 A 中元素的一种“计数”.)

集合 A 是**可数的**当且仅当存在某个函数将 A 一对一映射到自然数集 $\mathbb{N}$ 中. 例如，有限的集合显然是可数的. 现在来考虑可数无限集合 A，从给定的把 A 一对一映射到 $\mathbb{N}$ 中的 f，它可以扩展成从 A 到 $\mathbb{N}$ 上的一对一映射函数 f'. 对某个 $a_0 \in A$，如果 $f(a_0)$ 是值域 $\operatorname{ran} f$ 中的最小值，则令 $f'(a_0) = 0$. 一般地，存在唯一的 $a_n \in A$ 使得 $f(a_n)$ 是 $\operatorname{ran} f$ 中的第 $n+1$ 个元素，我们令 $f'(a_n) = n$. 注意 $A = \{a_0, a_1, \cdots\}$. (我们可以认为 f' 也是 A 中的元素的一种计数，只是计数过程是无限的.)

定理 0B 设 A 是一个可数集，则所有由 A 的元素组成的有限序列的集合也是可数的.

证明 所有有限序列的集合 S 可用如下式子表示：

$$S = \bigcup_{n \in \mathbb{N}} A^{n+1}.$$

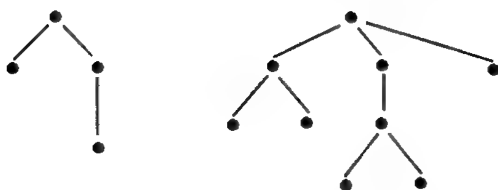
由于 A 是可数的，因此存在函数 f 将 A 一对一映射到 $\mathbb{N}$ 中.

基本思路是通过将 $\langle a_0, a_1, \cdots, a_m \rangle$ 指派给自然数 $2^{f(a_0)+1} 3^{f(a_1)+1} \cdot \cdots \cdot p_m^{f(a_m)+1}$ 来建立一个从 S 到 $\mathbb{N}$ 中的一对一映射，其中 p_m 是第 $m+1$ 个素数. 这样做的缺点是这种指派的定义不明确. 我们可以想像，可能会出现 $\langle a_0, a_1, \cdots, a_m \rangle = \langle b_0, b_1, \cdots, b_n \rangle$，且 a_i 和 b_j 属于 A，但是 $m \neq n$ 的情况. 然而，问题不大，只要给 S 中的每个元素指派具有上述形式的**最小**(smallest) 的一个数即可. 这就能够得到一个良定义的映射，并且容易看出它是一对一映射的. ■

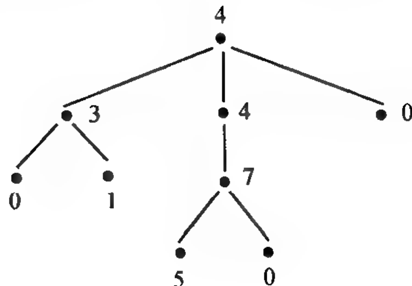
6

有些情况下，可能会用**树** (tree) 的方法来给出某些问题的直观图形表示. 可是，本书对树的说法是非正规的，定理和证明不会依赖于树. 相应地，在这里我们对树的讨论也是非正规的.

每棵树都是一种潜在的有限偏序，我们可以给出这种偏序关系 R 的图形表示；如果 $\langle a, b \rangle \in R$，那么我们把 a 放在 b 的下面，并且用线将它们连起来. 下面是两棵典型的树. (数学中的树是向下长的，而不是向上的.) 树中总会有一个最高点 (称之为**树根**(root)). 另外，尽管允许某个顶点向下分叉，一个顶点**上**面的点都在一条连线上.



除了这种向下的有限偏序树外，在一棵树上也可以定义一个标记函数，其定义域是顶点的集合. 比如，一棵树可以用自然数进行标记，如下图所示.



本书有些地方会用到选择公理. 但是, 当问题中所用的定理只涉及可数语言时, 我们可以不使用选择公理. 选择公理有很多等价的命题, 其中佐恩引理是非常有用的一个命题.

我们说一组集合 C 是一个 链(chain) 当且仅当对于 C 中任意的元素 x 和 y, 要么 $x \subseteq y$ 要么 $y \subseteq x$.

佐恩引理 设 A 是一个集合, 且满足对于任意的链 $C \subseteq A$, 有集合 $\bigcup C$ 在 A 中. 那么 A 中存在极大元 m, 即 m 不是 A 中其他任意元素的子集.

7

基数

所有的无限集合都很大, 但是其中某些集合会比另外一些更大. (例如, 实数集大于整数集.) 基数是一种度量集合大小的方便方法, 当然, 这种方法并非必不可少.

自然地, 我们说两个集合 A 与 B 大小相同, 当且仅当存在一个函数将 A 一对一地映射到 B 上. 如果 A 与 B 都是有限的, 这个概念就等价于通常的一个说法: 如果对 A 和 B 的元素分别计数, 那么得到的结果是相同的. 但是, 这个方法还可以用到当 A 和 B 是无限集合的时候, 尽管它们是难以计数的.

规范地说, A 与 B大小相等(记作 $A \sim B$) 当且仅当存在一个从 A 到 B 上的一对一映射. 例如, 自然数集 $\mathbb{N}$ 和整数集 $\mathbb{Z}$ 是大小相等的. 显然, 集合大小相等的关系是自反的、对称的和传递的.

对于有限集合, 我们可以使用自然数来度量其大小. 同一个自然数可以同时表示两个有限集合的大小, 当且仅当这两个集合大小相等. 基数的引入使我们可以将这种方法推广到无限集合.

一个集合 A 的 基数 (cardinal number)(或称为 基 (cardinality), 记作 card A) 是给集合 A 指定的一个特定的对象, 其意义为, 两个集合的基数相等当且仅当它们大小相等:

$$\text{card } A = \text{card } B \quad \text{iff} \quad A \sim B. \tag{K}$$

定义集合的基数的方法有很多种; 目前, 标准的方法是用与 A 大小相等的最小序数作为 card A. (这种定义的成功依赖于选择公理.) 这里并不讨论序数, 因为, card A 的实质是什么与我们的课程内容关系不大, 它并不比数字 2 的本质是什么有更多的意义. 对于我们来说, 最重要的是 (K) 式是成立的. 然而对于一个有限集合 A 而言, card A 指明了集合 A 中有多少个元素, 就已经够用了. 基数, 或简称 基, 就是指某个集合 A 的 card A.

(康托尔 (Georg Cantor) 于 1895 年最早采用基数这一概念, 他认为集合 M 的基数是从集合中抽象而得到的有助于人们理解的一个一般的概念, 而不考虑组成集合的元素的不同和集合中元素的排列次序.)

我们称 A 受控于 B (记作 $A \preccurlyeq B$) 当且仅当 A 与 B 的一个子集大小相等. 换句话说, $A \preccurlyeq B$ 当且仅当存在一个从 A 到 B 中的一对一映射. 对应的基数的关系是

$$\text{card } A \leqslant \text{card } B \;\; \text{iff} \;\; A \preccurlyeq B.$$

(易见, $\leqslant$ 是良定义的, 即 $\kappa \leqslant \lambda$ 是否成立仅取决于基数 κ 和 λ 本身, 而与集合的选择无关.) 受控性是自反的和传递的. 集合 A 受控于 $\mathbb{N}$ 当且仅当 A 是可数的. 下述定理是有关这方面内容的常用结果.

8

Schröder-Bernstein 定理 (a) 对于集合 A 和 B, 如果 $A \preccurlyeq B$ 并且 $B \preccurlyeq A$, 那么 $A \sim B$; (b) 对于任意的基数 κ 和 λ, 如果 $\kappa \leqslant \lambda$ 并且 $\lambda \leqslant \kappa$, 那么 $\kappa = \lambda$.

(b) 是从基数的角度对 (a) 的简单描述. 下面的定理等价于选择公理, 只是描述方式不同而已.

定理 0C (a) 对于任意的集合 A 和 B, $A \preccurlyeq B$ 或者 $B \preccurlyeq A$ 必有一个成立; (b) 对任意的基数 κ 和 λ, $\kappa \leqslant \lambda$ 或者 $\lambda \leqslant \kappa$ 必有一个成立.

这样, 任意两个基数都可以比较大小. (事实上, 任意非空的基数的集合都有一个最小的元素.) 最小的一类基数是有限集合的基数: 0, 1, 2, $\cdots$, 其次小的是无限基数 card $\mathbb{N}$, 称为 $\aleph_0$. 这样就有

$$0, 1, 2, \cdots, \aleph_0, \aleph_1, \cdots,$$

这里的 $\aleph_1$ 是大于 $\aleph_0$ 的最小基数. 实数的基数, card $\mathbb{R}$, 称作 "$2^{\aleph_0}$", 由于 $\mathbb{R}$ 是不可数的, 我们有 $\aleph_0 < 2^{\aleph_0}$.

常用于有限基数的加法与乘法运算也可以推广到所有的基数. 计算 $\kappa + \lambda$, 我们要选择两个不相交的集合 A 与 B, 其基数分别为 κ 和 λ, 那么,

$$\kappa + \lambda = \text{card}(A \cup B).$$

这个定义是良定义的, 即 $\kappa + \lambda$ 仅依赖于 κ 和 λ, 而不依赖于两个不相交集合 A 与 B 的选择. 对乘法, 有

$$\kappa \cdot \lambda = \text{card}(A \times B).$$

显然, 这些定义对有限基数都是正确的. 无限基数的算术运算都是相当简单的 (根据选择公理). 两个无限基数的和与积只是它们中较大的那个.

基数算术定理 对基数 κ 和 λ, 如果 $\kappa \leqslant \lambda$ 并且 λ 是无限的, 那么 $\kappa + \lambda = \lambda$. 另外, 如果 $\kappa \neq 0$, 那么 $\kappa \cdot \lambda = \lambda$. 特别地, 对于无限基数 κ,

9

$$\aleph_0 \cdot \kappa = \kappa.$$

定理 0D 对无限集合 A, 由 A 中元素组成的有限序列的集合 $\bigcup_n A^{n+1}$ 与 A 具有相同的基数 card A.

我们已经证明了这个定理对可数集合 A 是正确的 (参见定理 0B).

证明 根据基数算术定理 (使用 n 次), 每个 A^{n+1} 的基数等于 card A. 于是, 我们得到 $\aleph_0$ 个大小相等的集合的并集, 其基数为 $\aleph_0 \cdot \text{card } A = \text{card } A$. ■

例 代数数集合的基数为 $\aleph_0$. 首先, 我们把每个整系数的一元多项式看作其系数的序列; 然后, 由上述定理知, 共有 $\aleph_0$ 个多项式, 每个多项式都只有有限多个根. 为了给出一个充分大的上界, 注意到即使每个多项式都有 $\aleph_0$ 个根, 我们也只能得到 $\aleph_0 \cdot \aleph_0 = \aleph_0$ 个代数数. 同时, 至少也应该有这么多个, 于是结论成立.

由于实数是不可数的 (事实上, 有 $2^{\aleph_0}$ 个), 因此超越数也是不可数的 (也有 $2^{\aleph_0}$ 个).

10

第1章

命题逻辑

1.0 闲话形式语言

1.1 节我们要构建一种语言，可以将自然语言句子翻译成这种语言的形式. 与自然语言(譬如英语或者汉语) 不同的是，这种语言是一种具有精确格式要求的形式语言. 下面，在讲述这种精确格式之前先来考虑该语言的特点.

例如，句子“发现了钾的运动轨迹”在这种形式语言中可以翻译成符号 $\mathbf{K}$. 与之紧密相关的另外一个句子“未发现钾的运动轨迹”则可以用 $(\neg\ \mathbf{K})$ 来表达. 这里的 $\neg$ 是否定符号，读作“非”. 有人可能会将“未发现钾的运动轨迹”译作另外一个新的符号，比如 $\mathbf{J}$，但我们更愿意将这样的句子尽可能地分解为一些不能再分割的原子部分. 对于另外一个不相干的句子，“样本中含有氯”则可以译作符号 $\mathbf{C}$. 对下述的复合句，我们可以将它们翻译成右边的公式：

如果发现了钾的运动轨迹，那么样本中不含氯.	$(\mathbf{K} \to (\neg\ \mathbf{C}))$
样本中含有氯，且发现了钾的运动轨迹.	$(\mathbf{C} \wedge \mathbf{K})$

11

上面的第 2 个例子使用符号 $\wedge$ 来表示“且”，第 1 个例子使用了我们熟悉的箭头来表示“如果……，那么……”. 下面的例子使用符号 $\vee$ 来表示“或”：

没有发现钾的运动轨迹，或样本中不含氯.	$((\neg\ \mathbf{K}) \vee (\neg\ \mathbf{C}))$
样本中既不含氯也没有发现钾的运动轨迹.	$(\neg\ (\mathbf{C} \vee \mathbf{K}))$ 或 $((\neg\ \mathbf{C}) \wedge (\neg\ \mathbf{K}))$

最后一个句子我们有两种表示方法，它们之间的关系，稍后进行讨论.

复合句分解的一个重要的方面是：一旦我们给定了原子成分的真或假，就可以计算出整个复合句的真或假. 例如，假定化学家走出实验室，宣布她观察到了钾的运动轨迹，但是样本中不含氯. 那么我们就可以推断上述 4 个句子分别为真、假、真、假. 事实上，可以预先构造一个表格给出 4 种可能的实验结果 (如表 1-1 所示). 在 1.2 节中，我们将讨论这样的表格.

表 1-1

$\mathbf{K}$	$\mathbf{C}$	$(\neg\ (\mathbf{C} \vee \mathbf{K}))$	$((\neg\ \mathbf{C}) \wedge (\neg\ \mathbf{K}))$
F	F	T	T
F	T	F	F
T	F	F	F
T	T	F	F

使用形式语言可以避免自然语言的不精确性和模糊性，当然这不是没有代价的；形式语言的表达能力受到很大的限制.

为了描述一个形式语言，通常我们要给出以下 3 个信息.

(1) 指定使用的符号集 (字母表). 目前命题逻辑中使用的一些符号有

$$(,), \to, \neg, \mathbf{A}_1, \mathbf{A}_2, \cdots.$$

(2) 制定一些规则用以构造“语法正确”的有限符号串 (这样的符号串称为合式公式(well-formed formulas)). 例如，

12

$$(\mathbf{A}_1 \to (\neg \mathbf{A}_2))$$

是合式公式，而

$$)) \to \mathbf{A}_3$$

则不是.

(3) 要指明自然语言和形式语言之间所允许的翻译. 符号 $\mathbf{A}_1, \mathbf{A}_2, \cdots$ 是自然语言句子的翻译.

这里只有第 3 条给合式公式赋予了某种意义. 给公式指派意义的过程将引导我们进一步学习后续的内容. 但我们也会发现，从理论上说，在完全不知道合式公式的实际含义的前提下，仍然可以对合式公式进行各种各样的操作. 仅根据上述的前两条，我们也可以完成一些相关的操作，但这对我们来说并没有任何实际意义.

在继续学习之前，我们先简要地介绍一下另外一类现在被广泛关注的形式语言. 这类语言在数字计算机中得以应用 (至少与数字计算机有关联).

这类语言有很多种，在其中的一种语言中，

011010110101000111110001000001111010

是一个典型的合式公式；而

STEP#ADDIMAX, A

则是另外一种形式语言中的合式公式. (这里的 # 是空格符号，在字母表中引入这个符号是为了使得合式公式成为符号串.) **C++** 是一种应用广泛的程序设计语言，其中类似

while(*s + +);

这样的符号串都是合式公式.

在所有的情况中，都有一种给定方法将合式公式翻译为自然语言，并且将一类受限制的自然语言句子翻译为形式语言. 当然，计算机并不懂自然语言，它不能思考，只认识符号并机械地执行程序. 我们也可以用像计算机一样的方法来处理形式语言，虽然这个过程并不那么有趣.

1.1 命题逻辑的语言

假定我们给出了不同对象 (称之为符号) 的一个无限序列，这些符号在表 1-2 中给出其名称. 进一步假定，这些符号中没有一个是由其他符号的有限序列组成的.

13

表 1-2

符　号	名　称	描　述
(	左括号	标点
)	右括号	标点
$\neg$	否定符号	非
$\wedge$	合取符号	且
$\vee$	析取符号	或
$\rightarrow$	蕴涵符号	如果……，那么……
$\leftrightarrow$	等价符号	当且仅当
$\mathbf{A}_1$	第 1 个命题符号	
$\mathbf{A}_2$	第 2 个命题符号	
$\cdots$		
$\mathbf{A}_n$	第 n 个命题符号	
$\cdots$		

下面按序说明几个符号：

(1) $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ 这 5 个符号称为 *命题联结符*，上面已经给出了它们的汉语用法. 命题联结符和括号一起统称为 *逻辑符号*. 在形式语言与汉语的相互转换中，它们的作用是一致的. 而命题符号则称为 *参数* (或称为 *非逻辑符号*). 命题符号的翻译不是一成不变的；相反，它们可以有多种不同的解释，稍后我们会看到这一点.

(2) 命题符号有无穷多个. 一种比较合适的表示方法是使用一个命题符号 $\mathbf{A}$ 和一个撇号 $'$，这样我们就可以用潜在的无限序列

$$\mathbf{A}, \mathbf{A}', \mathbf{A}'', \cdots$$

取代

$$\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3, \cdots.$$

这种方法的优点是可以将不同的符号数目降低到 9 个. 另一种次优的方法是使用任意的命题符号集，不管可数与否. 本章中的大多数情况都使用后一种方法，只有 1.7 节例外.

[14] (3) 许多逻辑学家喜欢将 $\mathbf{A}_n$ 称为 *第n个命题* (proposition) 符号. 使用这一名称是由于要给句子 (sentence) 予一种特殊含义，要将命题 (proposition) 看成是一种肯定性的句子.

(4) 这些对象称为"符号". 从本体论的角度看，这些符号是中性的. 在上述列表的左边一列给出了符号的名称，例如 $\mathbf{A}_{243}$ 即第 243 个命题符号. (另外，$\mathbf{A}_{243}$ 也是符号的名称. 蕴涵符号具有形如箭头的几何性质，当然也可以不具有这样的性质，尽管它被称作"$\rightarrow$".) 这些符号本身可以是一些集合、数字、石子或者是语言学论域中的对象. 在最后一种情况中，这些符号可能就是一些名符其实的对象. 在第 2 章中，我们解释另外一种情况，命题符号本身可以是其他语言中的公式.

(5) 假定任何符号都不是其他符号的有限序列. 一方面，这些符号中任何一个要不同于其他符号 (如，$\mathbf{A}_3 \neq \leftrightarrow$)；另一方面它们也不是其他的两个或者更多符号构成的有限序列. 例如，要求 $\mathbf{A}_3 \neq \langle \neg, \mathbf{A}_4, (\rangle$. 这样规定的目的是为了保证符号的有限序列具有唯一分解性. 如果

$$\langle a_1, \cdots, a_m \rangle = \langle b_1, \cdots, b_n \rangle$$

且每个 a_i 和 b_j 是一个符号，那么 $m = n$ 且 $a_i = b_i$. (见第 0 章引理 0A 及相关注解.)

表达式 是符号的有限序列. 我们可以将一些符号连接构成一个表达式;$(\neg\ \mathbf{A}_1)$ 即由序列 $\langle (, \neg,\ \mathbf{A}_1,) \rangle$ 中的符号构成. 这种记法可以进行推广: 如果 α 与 β 是符号序列, 那么 $\alpha\beta$ 是由 α 与 β 按顺序连接而成的符号序列.

例如, 若 α, β 是下面两个等式给出的表达式:

$$\alpha = (\neg\ \mathbf{A}_1),$$

$$\beta = \mathbf{A}_2,$$

那么, $(\alpha \to \beta)$ 就是表达式

$$((\neg \mathbf{A}_1) \to\ \mathbf{A}_2).$$

下面来看一些例子, 这些例子都是把汉语句子翻译成形式语言. 设 $\mathbf{A}, \mathbf{B}, \cdots, \mathbf{Z}$ 是前 26 个命题符号 (如 $\mathbf{E} = \mathbf{A}_5$).

(1) 汉语: 这个嫌疑犯被释放. 翻译: $\mathbf{R}$

汉语: 得到的证据是可信的. 翻译: $\mathbf{E}$

汉语: 得到的证据不可信. 翻译: $(\neg\ \mathbf{E})$

15

汉语: 得到的证据是可信的, 嫌疑犯不能被释放. 翻译: $(\mathbf{E} \wedge (\neg\ \mathbf{R}))$

汉语: 要么得到的证据是可信的, 要么嫌疑犯被释放 (或者两者都可能). 翻译:: $(\mathbf{E} \vee \mathbf{R})$

汉语: 要么得到的证据是可信的, 要么嫌疑犯被释放 (两者只取其一). 翻译: $((\mathbf{E} \vee \mathbf{R}) \wedge (\neg(\mathbf{E} \wedge \mathbf{R})))$. 符号 $\vee$ 包含 "与/或" 的含义, 但通常表示 "或".

汉语: 得到的证据不可信, 但嫌疑犯不能被释放. 翻译: $((\neg\ \mathbf{E}) \wedge (\neg\ \mathbf{R}))$

汉语: 要么得到的证据不可信, 要么嫌疑犯不能被释放. 翻译: $((\neg\ \mathbf{E}) \vee (\neg\ \mathbf{R}))$

(2) 汉语: 如果愿望是马, 那么乞丐也可以驾驭. 翻译: $(\mathbf{W} \to \mathbf{B})$

汉语: 乞丐可以驾驭愿望当且仅当愿望是马. 翻译: $(\mathbf{B} \leftrightarrow \mathbf{W})$

(3) 汉语: 一件物品能够代表财富当且仅当它是可转让的、有限供应的、并且能够带来快乐或减轻痛苦. 翻译: $(\mathbf{W} \leftrightarrow (\mathbf{T} \wedge (\mathbf{L} \wedge (\mathbf{P} \vee \mathbf{Q}))))$, 这里 $\mathbf{W}$ 对应于 "一件物品能够代表财富." 在前面的例子中 $\mathbf{W}$ 对应的是另外一个不同的句子. 一个符号并不是只能用在一个翻译中.

需要注意的是: 不能将汉语中的句子 (如玫瑰花是红的) 与形式语言中的句子 (如 $\mathbf{R}$) 混淆起来. 它们的不同之处在于, 汉语句子一般有对错之分, 但在形式语言中的句子仅仅是一个符号序列. 虽然在某个上下文中, 一个形式化的句子能够表达一个正确的 (或错误的) 汉语句子, 但是在另外的上下文中它可能会有其他的解释.

有些表达式是没有意义的, 不能成为任何句子的翻译, 比如

$$((\to\ \mathbf{A}_3.$$

我们要将合式公式定义为 "语法正确" 的表达式, 就要去掉这些没有意义的表达式. 定义应该达到如下效果:

(a) 每个命题符号都是合式公式;

(b) 如果 α 和 β 是合式公式, 那么 $(\neg\alpha), (\alpha \wedge \beta), (\alpha \vee \beta), (\alpha \to \beta), (\alpha \leftrightarrow \beta)$ 也是合式公式;

(c) 只有通过上述 (a) 和 (b) 得到的表达式才是合式公式.

16

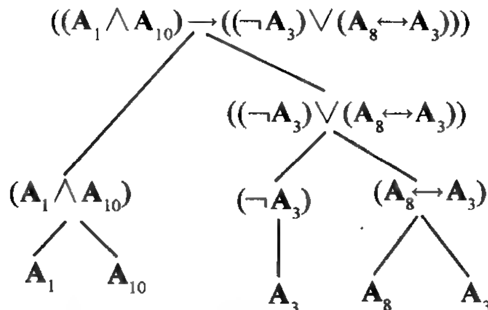
更准确地说，第 3 条性质应该表述为：合式公式(简称 公式) 是通过对命题符号有限次运用 构造公式的运算(formula-building operation) 而得到的表达式，构造公式的运算由下列的等式定义：

$$
\begin{aligned}
\mathcal{E}_{\neg}(\alpha) &= (\neg\alpha)\\
\mathcal{E}_{\wedge}(\alpha,\beta) &= (\alpha\wedge\beta),\\
\mathcal{E}_{\vee}(\alpha,\beta) &= (\alpha\vee\beta),\\
\mathcal{E}_{\rightarrow}(\alpha,\beta) &= (\alpha\rightarrow\beta),\\
\mathcal{E}_{\leftrightarrow}(\alpha,\beta) &= (\alpha\leftrightarrow\beta).
\end{aligned}
$$

例如，

$$((\mathbf{A}_1\wedge\mathbf{A}_{10})\rightarrow((\neg\mathbf{A}_3)\vee(\mathbf{A}_8\leftrightarrow\mathbf{A}_3)))$$

是一个合式公式，这可由下面的树看出：



从图中的生成树可以看出，表达式是用 4 个命题符号通过 5 次应用公式构造的运算得到的. 这个例子不典型，因为其中用到了所有的 5 个运算. 例如，A_3 是一个合式公式，它的生成树只是一个顶点，没有用到任何公式构造的运算. 如果“由命题符号构成”合式公式时，不考虑“空序列”的情况，则这是得到的不能再小的合式公式的例子了.

这类使用基本成分 (这里指命题符号) 和一些运算 (这里指 5 个运算) 构造的方法经常出现在逻辑和其他的数学分支中. 我们将在 1.4 节中以更一般的方式来考察这类构造方法.

17

下面详细描述“构造”的思想：如果对每个 $i\leqslant n$，构造序列 $\langle\varepsilon_1,\cdots,\varepsilon_n\rangle$ 至少满足以下三条中的一条，那么我们称 $\langle\varepsilon_1,\cdots,\varepsilon_n\rangle$ 是一个表达式的有限序列，

ε_i 是一个命题符号

对某个 $j<i$，使得 $\varepsilon_i=\mathcal{E}_{\neg}(\varepsilon_j)$；

对 $j<i,k<i$，使得 $\varepsilon_i=\mathcal{E}_{\square}(\varepsilon_j,\varepsilon_k)$.

其中 □ 是某个二元连接符号，即 $\wedge,\vee,\rightarrow,\leftrightarrow$ 中的一个. 那么，某个构造序列的最后一个表达式 α 就是合式公式. 在构造过程中，我们将 ε_i 看作第 i 个阶段形成的表达式.

对于前面的例子，

$$((\mathbf{A}_1\wedge\mathbf{A}_{10})\rightarrow((\neg\mathbf{A}_3)\vee(\mathbf{A}_8\leftrightarrow\mathbf{A}_3)))$$

将其生成树压缩成为一个线性序列，我们就可以得到它的一个构造序列.

这类构造的一个特点是服从 归纳法则(induction principle). 称集合 S 在二元函数 f 作用下是 封闭的(close), 当且仅当无论何时对于任意的 $x\in S$ 和 $y\in S$ 都有 $f(x,y)\in S$，我们可以类似地定义一元函数的封闭性, 等等.

归纳法则 如果 S 是一个包含所有命题符号的合式公式的集合，并且在 5 种运算的作用下是封闭的，那么 S 是所有合式公式的集合.

证明 1 考虑任意的合式公式 α，它是由命题符号有限次使用公式构造运算得到的. 观察它所对应的生成树，我们可以看到树中每个表达式都属于 S. 最后 (即经过有限步后)，我们会在树的顶部发现 $\alpha \in S$. ■

证明 2 不使用树来证明这个原理. 考虑任意的一个合式公式 α，α 是某个构造序列的 $\langle \varepsilon_1, \cdots, \varepsilon_n \rangle$ 的最后一个成员. 我们对 i 使用通常的强数学归纳法，要证明对于每个 $i \leqslant n$ 都有 $\varepsilon_i \in S$.

即按照归纳假设，假设对于每个 $j < i$ 都有 $\varepsilon_j \in S$，那么对各种情况都加以考虑，我们就能得出 $\varepsilon_i \in S$. 因此，根据对 i 的强归纳法，我们可以得到对每个 $i \leqslant n$ 都有 $\varepsilon_i \in S$. 特别地，最后一个元素 α 属于 S. ■

接下来的学习中，我们经常要使用这个原理. 下面的例子就是用这个原理来证明一个表达式不是合式公式.

18

例 一个表达式中如果左括号比右括号多，那么它不是合式公式.

证明 我们从一个没有左括号和右括号的命题符号开始，应用公式构造运算，而每次应用公式构造运算都会增加一对匹配的括号. 我们重新描述一下这个思路："平衡" 的合式公式 (即含有相同数目的左括号和右括号) 的集合包含所有的命题符号，并且在公式构造运算作用下是封闭的. 那么，归纳法则就保证了所有的合式公式是平衡的. ■

公式构造运算的一个特性是 *向上* 构造而不是 *向下* 的，即表达式 $\mathcal{E}_\square(\alpha, \beta)$ 总是包含 α 和 β 的完整序列作为其组成部分，然后再加上其他的一些符号. 特别地，这个表达式比 α 或 β 都长.

一个给定的合式公式 φ 是 *怎样* 精确构成的？这个问题可以通过上述的特性得以简化. 可以说，所有的构造部件都是序列 φ 的子段. 例如，如果 φ 不含有符号 $\mathbf{A}_4$，那么构造过程也不会用到 $\mathbf{A}_4$. (见习题 4.)

习题

1. 给出 3 个句子并翻译成形式语言. 要选择有趣的句子在结构，以使得每个句子在翻译成形式语言后都包括 15 个以上的符号.
2. 证明不存在长度为 2、3、6 的合式公式，但其他任意正整数长度的合式公式都可能存在.
3. 设 α 是一个合式公式，c 是二元连接符 $(\wedge, \vee, \rightarrow, \leftrightarrow)$ 出现的次数，s 是命题符号出现的次数 (例如，若 α 是 $(\mathbf{A} \rightarrow (\neg \mathbf{A}))$，则 $c = 1, s = 2$). 使用归纳法则证明 $s = c + 1$.
4. 假定一个构造序列的最后一个成员是 φ，且 φ 不包含符号 $\mathbf{A}_4$，假设在构造序列中删除所有含有 $\mathbf{A}_4$ 的表达式，证明其结果仍是一个合法的构造序列.
5. 设 α 是一个不包含否定符号 $\neg$ 的合式公式.
   (a) 证明 α 的长度是奇数 (长度是指 α 中所有出现的符号的个数).
   (b) 证明出现在 α 中的命题符号数超过所有符号数的 1/4.
   提示：用归纳法证明 α 的总长度是 $4k + 1$ 且命题符号数至少是 $k + 1$.

19

1.2　真值指派

我们希望定义，在形式语言中一个合式公式的含义，在逻辑上如何从其他的合式公式得到．例如，$\mathbf{A}_1$ 的含义可以从 $(\mathbf{A}_1 \wedge \mathbf{A}_2)$ 得到，无论参数 $\mathbf{A}_1$ 和 $\mathbf{A}_2$ 在汉语中的具体含义是什么，只要 $(\mathbf{A}_1 \wedge \mathbf{A}_2)$ 是真的，那么 $\mathbf{A}_1$ 也是真的．当然，$\mathbf{A}_1$ 这个符号翻译成汉语，会有多种不同的译法，因此其意义也可能是模糊的．幸运的是，有一种简单而准确的方法来表达翻译这个概念的本质．

真值(truth value) 集合 $\{F,T\}$ 包含了两个不同的值：

F：假；
T：真．

(这两个值本身是什么无关紧要，它们也可以记作 0 和 1．) 对于命题符号集合 S，一个 **真值指派**(truth assignment)v 是指一个函数

$$v : S \to \{F,T\}.$$

这个函数给 S 中的每个符号指定一个真值：T 或 F．真值指派在下面的讨论中取代了形式语言到汉语的翻译．

(这里我们主要讨论 **二值** 逻辑．当然，三值逻辑也是可以研究的，那样会有一个具有三个值的真值集合．进一步说，也可以允许使用 512 个真值或者 $\aleph_0$ 个真值；甚或可以使用区间 [0,1] 或者其他方便的值空间作为真值集合．一个非常有意义的特例是真值取自于最有意义的布尔代数，这是一个非常重要的二值逻辑，我们将只讨论这种情形．)

设 $\bar{S}$ 是由 S 通过 5 种公式构造运算得到的合式公式的集合．($\bar{S}$ 也可以看作所有命题逻辑符全部来自 S 的合式公式的集合；参见下一节结尾的注释．) 我们可以将 v 扩展到 $\bar{v}$,

$$\bar{v} : \bar{S} \to \{F,T\},$$

$\bar{v}$ 给 $\bar{S}$ 中的每个合式公式指派一个真值，要求满足以下条件：

(0) 对任意的 $A \in S$，$\bar{v}(A) = v(A)$．(因此称 $\bar{v}$ 为 v 的扩展．) 对于 $\bar{S}$ 中任意的 α 和 β,

20

(1)
$$\bar{v}((\neg\alpha)) = \begin{cases} T & \text{如果 } \bar{v}(\alpha) = F \\ F & \text{其他情况,} \end{cases}$$

(2)
$$\bar{v}((\alpha \wedge \beta)) = \begin{cases} T & \text{如果 } \bar{v}(\alpha) = T \text{ 且 } \bar{v}(\beta) = T \\ F & \text{其他情况,} \end{cases}$$

(3)
$$\bar{v}((\alpha \vee \beta)) = \begin{cases} T & \text{如果 } \bar{v}(\alpha) = T \text{ 或 } \bar{v}(\beta) = T \text{ (或两者成立)} \\ F & \text{其他情况,} \end{cases}$$

(4)
$$\bar{v}((\alpha \to \beta)) = \begin{cases} F & \text{如果 } \bar{v}(\alpha) = T \text{ 且 } \bar{v}(\beta) = F \\ T & \text{其他情况.} \end{cases}$$

$$(5) \qquad \bar{v}\,((\alpha \leftrightarrow \beta)) = \begin{cases} T & \text{如果 } \bar{v}(\alpha) = \bar{v}(\beta) \\ F & \text{其他情况.} \end{cases}$$

条件 1~5 构成了表 1-3. 从这里开始, 形式语言开始使用合取符号的含义. 特别要注意的是 $\rightarrow$ 的含义, 无论何时 α 的赋值为 F, 那么 $(\alpha \rightarrow \beta)$ 就 "毋庸置疑是真的", 就被赋以真值 T. 读者可能会问, 这些符号能否准确反映我们日常生活中说的 "如果……, 那么……" "或者" 等. 但我们更为关注的是数学表达而不是日常说法的准确性.

表 1-3

α	β	$(\neg\alpha)$	$(\alpha\wedge\beta)$	$(\alpha\vee\beta)$	$(\alpha\rightarrow\beta)$	$(\alpha\leftrightarrow\beta)$
T	T	F	T	T	T	T
T	F	F	F	T	F	F
F	T	T	F	T	T	F
F	F	T	F	F	T	T

例如, 汉语句子 "如果你说的是真的, 我就是猴子的叔叔. " 可以用形式语言中的公式 $(\mathbf{V} \rightarrow \mathbf{M})$ 来表达. 只要你撒谎, 我们就给这个公式赋以真值 $\mathbf{T}$. 这时, 我们当然不是说, 你说实话和我有一个猴子侄子之间会存在任何的因果关系. 这个句子是一个条件句, 关于我侄子的断言需要一个特定的条件——你是不是说了实话. 只要这个条件不对, 句子本身就毋庸置疑是真的.

大致地说, 我们可以把条件公式 $(\alpha \rightarrow \beta)$ 看成一个*承诺*(promise): 只要特定条件满足了 (即 α 是真的), 那么 β 也是真的. 如果条件 α 不能满足, 不管 β 的真假, 这个承诺都有效.

21

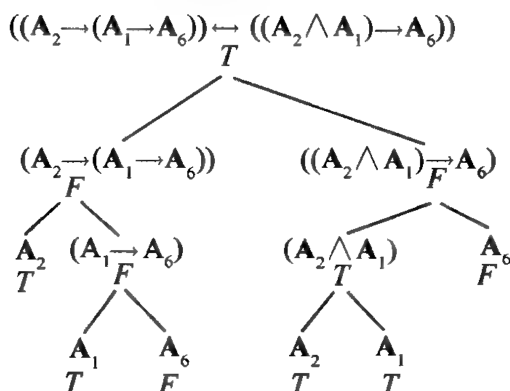
看一个计算 $\bar{v}$ 的例子, 设 α 是一个合式公式

$$((\mathbf{A}_2 \rightarrow (\mathbf{A}_1 \rightarrow \mathbf{A}_6)) \leftrightarrow ((\mathbf{A}_2 \wedge \mathbf{A}_1) \rightarrow \mathbf{A}_6)).$$

设真值指派 v 给 $\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_6\}$ 的指派如下:

$$v(\mathbf{A}_1) = T,$$
$$v(\mathbf{A}_2) = T,$$
$$v(\mathbf{A}_6) = F.$$

我们来计算 $\bar{v}(\alpha)$. 首先看下面给出的 α 的生成树.

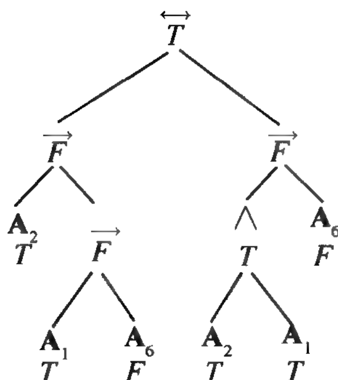


从树的底部开始，给树中的每一个顶点 β 赋真值 $\bar{v}(\beta)$，因此第 1 步计算

$$\bar{v}((\mathbf{A}_1 \to \mathbf{A}_6)) = F \text{ 且 } \bar{v}((\mathbf{A}_2 \wedge \mathbf{A}_1)) = T.$$

接下来，我们计算 $\bar{v}((\mathbf{A}_2 \to (\mathbf{A}_1 \to \mathbf{A}_6))) = F$，依此类推，最终可以得到 $\bar{v}(\alpha) = T$.

事实上，这个计算可以通过一个更为简洁的方式完成. 首先，树可以直接使用真值作为顶点：



22

甚至，这个计算还可以简缩为 (带括号的) 一行：

$$\begin{array}{c}((\mathbf{A}_2 \to (\mathbf{A}_1 \to \mathbf{A}_6)) \leftrightarrow ((\mathbf{A}_2 \wedge \mathbf{A}_1) \to \mathbf{A}_6)).\\ T\ \ F\ \ T\ \ FF\ \ \ T\ \ \ TTT\ \ F\ \ F\end{array}$$

定理 12A 对于集合 S 的任意真值指派 v，存在唯一的函数 $\bar{v}: \bar{S} \to \{F, T\}$ 满足前述的条件 0~5.

定理的全部证明将会在 1.3 节和 1.4 节中给出，不过，由前面的例子，我们已经可以看出这个定理的合理性. 在证明 $\bar{v}$ 的存在性时，实际上最关键的是要证明前面例子中提到的生成树的唯一性.

我们称一个真值指派 v *满足*(satisfy)φ 当且仅当 $\bar{v}(\varphi) = T$. (当然，在这种情况下，v 的赋值范围必须包括 φ 中出现的所有命题符号.) 现在考虑一个合式公式的集合 (看作是假设前提) 和另一个合式公式 τ(看作是可能的结论).

定义 Σ *重言蕴涵*(tautologically imply)τ(记作 $\Sigma \models \tau$)，当且仅当满足 Σ 中每个合式公式的真值指派也满足 τ.

该定义反映了这样一个直觉：一个结论可以从一些假设条件推出，是指在假设条件是正确的前提下，得到的结论也是正确的.

值得我们关注的是重言蕴涵这个概念的几个特例. 第一个是，Σ 取空集 $\varnothing$. 请注意，任意的真值指派都能够满足 $\varnothing$ 中的元素 ($\varnothing$ 中没有元素，当然这是正确的)，因此我们有 $\varnothing \models \tau$ 当且仅当每个真值指派 (要求对 τ 中出现的所有命题符号都有指派) 都满足 τ. 这时我们称 τ 是*重言式*(tautology)，记作 $\models \tau$. 比如，在刚才的例子中，我们看到对 $\{\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_6\}$ 的一个真值指派能够满足合式公式 $((\mathbf{A}_2 \to (\mathbf{A}_1 \to \mathbf{A}_6)) \leftrightarrow ((\mathbf{A}_2 \wedge \mathbf{A}_1) \to \mathbf{A}_6))$. 事实上，其他的 7 个指派也都能满足该合式公式，所以这是一个重言式.

另外一个特例是，任意一个真值指派都不能完全满足 Σ 中所有合式公式. 这时，对任意的 τ, $\Sigma \models \tau$ 也必然是真的. 例如，

$$\{\mathbf{A}, (\neg\mathbf{A})\} \models \mathbf{B}.$$

这些结果中没有更深的原理，都是我们前面定义的附带结果而已.

例 $\{\mathbf{A}, (\mathbf{A} \rightarrow \mathbf{B})\} \models \mathbf{B}$. 对于 $\{\mathbf{A}, \mathbf{B}\}$，我们有 4 个真值指派. 容易验证，满足 $\mathbf{A}$ 和 $(\mathbf{A} \rightarrow \mathbf{B})$ 的任意指派 v 都有 $v(\mathbf{A}) = v(\mathbf{B}) = T$. 这样，$v$ 同样也满足 $\mathbf{B}$.

23

如果 Σ 中只有一个元素 σ, 我们用记法 "$\sigma \models \tau$" 取代 "$\{\sigma\} \models \tau$". 如果 $\sigma \models \tau$ 和 $\tau \models \sigma$ 同时成立，那么 σ 和 τ 称为 *重言等价的*(tautologically equivalent)(记作 $\sigma \models\!\dashv \tau$). 例如，在 1.0 节中我们认为合式公式 $(\neg\, (\mathbf{C} \vee \mathbf{K}))$ 和 $((\neg\, \mathbf{C}) \wedge (\neg\, \mathbf{K}))$ 是同一个句子的表述，我们可以验证二者是重言等价的.

下面的定理我们将在 1.7 节中给出证明.

紧致性定理 设 Σ 是合式公式的无限集合，如果对于 Σ 中的任意有限子集 Σ_0，都存在一个真值指派满足 Σ_0 中的每个合式公式，那么，就存在一个真值指派满足 Σ 的所有合式公式.

这个定理也可以简单地描述成：如果 Σ 的每个有限子集都是可满足的，那么 Σ 本身也是可满足的. (熟悉拓扑学的读者可以考察称其为"紧致性定理"的原因，这确实是一个特定拓扑空间的紧致性. 在积空间上使用吉洪诺夫定理可以证明这个定理.)

1.2.1 真值表

下面我们给出一个系统的过程，对于给定的合式公式 $\sigma_1, \sigma_2, \cdots, \sigma_k$ 和 τ，来验证 $\{\sigma_1, \sigma_2, \cdots, \sigma_k\} \models \tau$ 是否成立.

特别地，当 $k = 0$ 时，就变成验证给定的合式公式是否是重言式了.

作为第一个例子，证明

$$(\neg\, (\mathbf{A} \wedge \mathbf{B})) \models ((\neg\, \mathbf{A}) \vee (\neg\, \mathbf{B})).$$

为了证明，考察 $\{\mathbf{A}, \mathbf{B}\}$ 所有的 4 个真值指派，如下表所示 (一般地，n 个命题符号的真值指派的个数共有 2^n 个).

A	**B**
T	T
T	F
F	T
F	F

这个表可以加上 $(\neg(\mathbf{A} \wedge \mathbf{B}))$ 和 $((\neg\mathbf{A}) \vee (\neg\mathbf{B}))$ 两列，并分别计算这两个合式公式的真假值，写在相应的位置上 (见表 1-4). (表 1-4 中最左边的两列实际上是无需列出的.) 可以看出，所有满足 $(\neg(\mathbf{A} \wedge \mathbf{B}))$ 的真值指派 (共有 3 个) 也满足 $((\neg\mathbf{A}) \vee (\neg\mathbf{B}))$，反之亦然.

24

这样，我们就有 $(\neg(\mathbf{A} \wedge \mathbf{B})) \models\!\dashv ((\neg\mathbf{A}) \vee (\neg\mathbf{B}))$.

用同样的方法可以证明 $(\neg(\mathbf{A}\wedge\mathbf{B}))\nvDash((\neg\mathbf{A})\wedge(\neg\mathbf{B}))$. 实际上，只要在表中有一行说明存在一个真值指派满足 $(\neg(\mathbf{A}\wedge\mathbf{B}))$ 而不能满足 $((\neg\mathbf{A})\wedge(\neg\mathbf{B}))$ 就足够了.

一般来说，这个方法效率不高. 比如，要证明

$$\vDash((\mathbf{A}\vee(\mathbf{B}\wedge\mathbf{C}))\leftrightarrow((\mathbf{A}\vee\mathbf{B})\wedge(\mathbf{A}\vee\mathbf{C}))),$$

我们仍可以使用真值表，不过，这一次对 $\{\mathbf{A},\mathbf{B},\mathbf{C}\}$ 的真值指派共有 8 个，所以真值表就需要有 8 行. 简单地改进一下，这个繁琐的过程可以简化为

$$((\mathbf{A}\vee(\mathbf{B}\wedge\mathbf{C}))\leftrightarrow((\mathbf{A}\vee\mathbf{B})\wedge(\mathbf{A}\vee\mathbf{C}))).$$

$T\ T\qquad\quad T\ \ T\ T\quad\ T\ T\ T$

$F\ F\ F\ F\quad\ T\ \ F\ F\ F\ F\ \ F$

$F\ T\ T\ T\ T\ T\ \ F\ T\ T\ T\ \ F\ T\ T$

表　1-4

A	**B**	$(\neg\,(\mathbf{A}\wedge\mathbf{B}))$	$((\neg\,\mathbf{A})\vee(\neg\,\mathbf{B}))$
T	T	$FTTT$	$FTFFT$
T	F	$TTFF$	$FTTTF$
F	T	$TFFT$	$TFTFT$
F	F	$TFFF$	$TFTTF$

在第 1 行中，假定只有 $v(\mathbf{A})=T$ 就够了，因为对于这个合式公式已经提供了足够的条件使其真值为 T，接下来的各行假定 $v(\mathbf{A})=F$. 并且第 2 行假定 $v(\mathbf{B})=F$, 这也能够得出这个合式公式的真值为 T. 由于在该表达式中 $\mathbf{B}$ 与 $\mathbf{C}$ 是对称的，所以我们只需再假定 $v(\mathbf{C})=T$, 这就给出了第 3 行，到此该过程结束.

验证下面的重言式，我们使用上述的方法就不用使用 16 行的真值表了.

$$((((\mathbf{P}\wedge\mathbf{Q})\rightarrow\mathbf{R})\rightarrow\mathbf{S})\rightarrow((\mathbf{P}\rightarrow\mathbf{R})\rightarrow\mathbf{S})).$$

$\qquad\qquad\qquad\qquad\qquad T\qquad\quad\ \ T\ T$

$F\qquad T\quad F\ F\quad T$

$T\ T\ T\quad R\ R\ \bar{R}\ F\quad T\quad T\ R\ R\ \bar{R}\ F$

[25] 这里的第 1 行处理了 $v(\mathbf{S})=T$ 的情形，第 2 行处理了 $v(\mathbf{P})=F$ 或者 $v(\mathbf{Q})=F$ 的情形，第 3 行合并了其他各种可能性，这里的 R 是指派给 $\mathbf{R}$ 的一个真值，$\bar{R}$ 取与 R 相反的值.

显而易见，上面例子中的公式是重言式，它的前提条件 (表达式的左边) 很强，而结论则相对比较弱. 这样，

$$(\mathbf{P}\wedge\mathbf{Q})\vDash\mathbf{P},$$
$$(\mathbf{P}\rightarrow\mathbf{R})\vDash((\mathbf{P}\wedge\mathbf{Q})\rightarrow\mathbf{R}),$$
$$(((\mathbf{P}\wedge\mathbf{Q})\rightarrow\mathbf{R})\rightarrow\mathbf{S})\vDash((\mathbf{P}\rightarrow\mathbf{R})\rightarrow\mathbf{S}).$$

开发有效的方法来简化繁琐的方法，对于用机器证明定理来说是非常重要的. 有些程序要求检验命题逻辑中的合式公式，而这些合式公式往往会包含上千个命题符号，如果使用真值表的方法就太繁琐了. 因此，如何开发高效的方法已经成为当今计算机科学研究的一个领域.

附注 如果对一个包含 n 个命题符号的合式公式使用完整的真值表，其真值表有 2^n 行. 随着 n 的增长，2^n 是以“指数”增长的，这样的增长速度是非常难以处理的. 譬如，假定以 1 000 000 行每秒的速度生成真值表 (借助计算机)，当 n=80 时，那就需要 2^{80}μs 生成真值表，这个时间是多长呢？换算成以年为单位的话，大概是 380 亿年，而宇宙至今也只存在了大概 150 亿年. 结论就是 2^{80}μs 超过了宇宙到现在的存在时间！

有没有快一些的方法呢？对于给定的含有 n 个命题符号的合式公式 α，确定它是不是重言式，有没有只需要 $10n^5$μs 左右的一般方法？(或者其他的 n 的多项式函数时间，而不是指数函数). (如果真有这样的方法，那么当 n=80 时，$10n^5$ μs 也不过就是 9 h.) 这些问题的答案是否存在不得而知，然而大家普遍认为这样的方法是不可能存在的. 此类问题被称为“P 与 NP 问题”，是现代理论计算机科学中最著名的难题.

1.2.2 典型的重言式

26

(1) ∧、∨、↔ 的结合律和交换律.

(2) 分配律：

$$((\mathbf{A} \wedge (\mathbf{B} \vee \mathbf{C})) \leftrightarrow ((\mathbf{A} \wedge \mathbf{B}) \vee (\mathbf{A} \wedge \mathbf{C}))).$$
$$((\mathbf{A} \vee (\mathbf{B} \wedge \mathbf{C})) \leftrightarrow ((\mathbf{A} \vee \mathbf{B}) \wedge (\mathbf{A} \vee \mathbf{C}))).$$

(3) 否定：

$$((\neg(\neg \mathbf{A})) \leftrightarrow \mathbf{A}).$$
$$((\neg(\mathbf{A} \to \mathbf{B})) \leftrightarrow (\mathbf{A} \wedge (\neg \mathbf{B}))).$$
$$((\neg(\mathbf{A} \leftrightarrow \mathbf{B})) \leftrightarrow ((\mathbf{A} \wedge (\neg \mathbf{B})) \vee ((\neg \mathbf{A}) \wedge \mathbf{B}))).$$

德摩根律：

$$((\neg(\mathbf{A} \wedge \mathbf{B})) \leftrightarrow ((\neg \mathbf{A}) \vee (\neg \mathbf{B}))).$$
$$((\neg(\mathbf{A} \vee \mathbf{B})) \leftrightarrow ((\neg \mathbf{A}) \wedge (\neg \mathbf{B}))).$$

(4) 其他.

排中律：$(\mathbf{A} \vee (\neg \mathbf{A}))$.
矛盾律：$(\neg(\mathbf{A} \wedge (\neg \mathbf{A})))$.
逆否律：$((\mathbf{A} \to \mathbf{B}) \leftrightarrow ((\neg \mathbf{B}) \to (\neg \mathbf{A})))$.
输出律：$(((\mathbf{A} \wedge \mathbf{B}) \to \mathbf{C}) \leftrightarrow (\mathbf{A} \to (\mathbf{B} \to \mathbf{C})))$.

习题

1. 证明下列两个公式中的任何一个都不能够重言蕴涵另一个：

$$(\mathbf{A} \leftrightarrow (\mathbf{B} \leftrightarrow \mathbf{C})),$$
$$((\mathbf{A} \wedge (\mathbf{B} \wedge \mathbf{C})) \vee ((\neg \mathbf{A}) \wedge ((\neg \mathbf{B}) \wedge (\neg \mathbf{C}))))$$

提示：只需要两个真值指派，而不是 8 个.

2. (a) $(((\mathbf{P} \to \mathbf{Q}) \to \mathbf{P}) \to \mathbf{P})$ 是重言式吗？
   (b) 递归定义 σ_k 如下：$\sigma_0 = (\mathbf{P} \to \mathbf{Q}), \sigma_{k+1} = (\sigma_k \to \mathbf{P})$. 试确定对哪些 k, σ_k 是重言式？((a) 对应的是 k=2 的情形.)

3. (a) $((\mathbf{P} \to \mathbf{Q}) \vee (\mathbf{Q} \to \mathbf{P}))$ 是否是重言式？
   (b) $((\mathbf{P} \wedge \mathbf{Q}) \to \mathbf{R})$ 是不是重言蕴涵了 $((\mathbf{P} \to \mathbf{R}) \vee (\mathbf{Q} \to \mathbf{R}))$？
4. 证明
   (a) $\Sigma; \alpha \models \beta$ 当且仅当 $\Sigma \models (\alpha \to \beta)$.
   (b) $\alpha \models\!\dashv \beta$ 当且仅当 $\models (\alpha \leftrightarrow \beta)$.
   (提示：$\Sigma;\ \alpha = \Sigma \bigcup \{\alpha\}$ 是 Σ 加上新元素 α 构成的集合.)
5. 证明或者反驳下列断言：
   (a) 如果 $\Sigma \models \alpha$ 或者 $\Sigma \models \beta$, 那么有 $\Sigma \models (\alpha \vee \beta)$.
   (b) 如果 $\Sigma \models (\alpha \vee \beta)$, 那么 $\Sigma \models \alpha$ 或者 $\Sigma \models \beta$.
6. 证明
   (a) 如果两个真值指派 v_1 和 v_2 与在合式公式 α 中出现的命题符号的指派是一致的, 那么 $\bar{v}_1(\alpha) = \bar{v}_2(\alpha)$. (使用归纳法则.)

27

   (b) 设 S 是包含在 Σ 和 τ 中出现的命题符号的集合. 试证明 $\Sigma \models \tau$ 当且仅当 S 的每个真值指派只要满足 Σ 中的每个元素也就满足 τ.
   (由 (a) 很容易得到这个结果, (b) 的意义在于：我们无需担心一个真值指派是否为所有命题符号都给出了赋值, 而只要对用得着的命题符号赋了值就够了. 例如, 想要对所有命题符号给出真值指派, 难在可能有无限多个命题符号, 甚至可能有不可数个命题符号.)
7. 假定你居住在某个岛上, 岛上的居民总是说真话或者总是说假话. 一天, 你走到一个岔路口想要知道哪条路通往城镇. 恰好有一个当地居民路过, 但是他只有时间回答一个是或者不是. 你需要问一个什么样的问题才能知道该走哪条路？提示：造表.
8. (置换)$\alpha_1, \alpha_2, \cdots$ 一个合式公式的序列, 对每个合式公式 φ, 令 φ^* 是用 α_n 置换 $\mathbf{A}_n$(对所有 n) 后得到的结果.
   (a) 设 v 是所有命题符号的真值指派, 定义 u 为满足 $u(\mathbf{A}_n) = \bar{v}(\alpha_n)$ 的真值指派, 证明 $\bar{u}(\varphi) = \bar{v}(\varphi^*)$ 使用归纳法.
   (b) 证明如果 φ 是重言式, 那么 φ^* 也是. (例如, $((\mathbf{A} \wedge \mathbf{B}) \leftrightarrow (\mathbf{B} \wedge \mathbf{A}))$ 是一个重言式, 因此对任意的合式公式 α、β, 通过置换可得 $((\alpha \wedge \beta) \leftrightarrow (\beta \wedge \alpha))$ 是重言式.)
9. (对偶性) 设 α 是一个合式公式, 其连接符号为 $\wedge$、$\vee$、$\neg$. 设 α^* 是将 α 中 $\wedge$ 与 $\vee$ 对换, 同时将每个命题符号用其否定代替. 证明 α^* 重言等价于 $(\neg\alpha)$. 使用归纳法.
   说明：这意味着如果 $\alpha \models\!\dashv \beta$ 那么 $\alpha^* \models\!\dashv \beta^*$.
10. 我们说合式公式集合 Σ_1 与 Σ_2 是*等价的*, 当且仅当对于任意的合式公式 α 有 $\Sigma_1 \models \alpha$ 当且仅当 $\Sigma_2 \models \alpha$. 称合式公式集合 Σ 是*独立的*当且仅当其中任意一个合式公式都不能被 Σ 中的其他合式公式重言蕴涵. 试证明以下结论：
    (a) 有限的合式公式集合都存在一个独立的等价子集.
    (b) 无限的合式公式集合不一定存在独立的等价子集.
    *(c) 令 $\Sigma = \{\sigma_1, \sigma_2, \cdots\}$; 证明存在独立的等价集合 Σ'. (由 (b) 可知, 一般情况, $\Sigma' \subseteq \Sigma$ 是不正确的.)

28

11. 证明真值指派 v 满足合式公式

$$(\cdots(\mathbf{A}_1 \leftrightarrow \mathbf{A}_2) \leftrightarrow \cdots \leftrightarrow \mathbf{A}_n)$$

    当且仅当有偶数个 $i (1 \leqslant i \leqslant n)$, 使得 $v(\mathbf{A}_i) = F$. (由 $\leftrightarrow$ 的结合律, 括号的位置无关紧要.)
12. 有 3 个杀人嫌疑犯：Adams、Brown 和 Clark. Adams 说 “不是我杀的. 死者是 Brown 的老熟人. Clark 仇恨他. ” Brown 说 “不是我杀的. 我根本不认识死者, 而且案发的那个星期我不在本地. ” Clark 说 “不是我杀的. 那天我在市区看见 Adams、Brown 和死者在一起, 必定是他们中的一个人干的. ” 如果无辜的人说的都是实话, 而杀人犯可能不说实话. 请问谁是杀人犯？

13. 网球杂志的一则广告宣称:“如果我不在打网球，那么我就在看网球. 如果我不在看网球，那么我就在看网球杂志. ”假定说这话的人不能同时做两件以上的事，那么他在做什么?(翻译成形式语言，考虑可能的真值指派.)
14. 设 S 是所有命题符号的集合，假定 $v: S \to \{F, T\}$ 是真值指派. 证明**最多**存在一个 v 的扩展 $\bar{v}$ 满足本节开始所列的 0~5 的条件. (假定有两个扩展 $\bar{v}_1$, $\bar{v}_2$, 使用归纳法证明两个扩展相等.)
15. 下面三个公式中，哪个能够重言蕴涵另外一个?
    (a) $(\mathbf{A} \leftrightarrow \mathbf{B})$.
    (b) $(\neg((\mathbf{A} \to \mathbf{B}) \to (\neg(\mathbf{B} \to \mathbf{A}))))$.
    (c) $(((\neg\mathbf{A}) \vee \mathbf{B}) \wedge (\mathbf{A} \vee (\neg\mathbf{B})))$.

1.3 解析算法

在分析合式公式时，我们使用括号来消除其语义的模糊性，本节的目的是证明这样做的正确性. (作为真值指派 v 的扩展 $\bar{v}$ 是否存在取决于是否能够消除其模糊性 [1].)

首先考虑的是不包括括号的情况. 我们用下面的合式公式作为例子来说明其模糊性:

$$\mathbf{A}_1 \vee \mathbf{A}_2 \wedge \mathbf{A}_3,$$

29

这个公式对应于两种不同的构成形式，即 $((\mathbf{A}_1 \vee \mathbf{A}_2) \wedge \mathbf{A}_3)$ 和 $(\mathbf{A}_1 \vee (\mathbf{A}_2 \wedge \mathbf{A}_3))$. 如 $v(\mathbf{A}_1) = T, v(\mathbf{A}_3) = F$, 那么在计算 $\bar{v}(\mathbf{A}_1 \vee \mathbf{A}_2 \wedge \mathbf{A}_3)$ 时就会出现矛盾.

我们要证明在使用括号后这种模糊性是可以消除的，并且每个合式公式有唯一的构成形式. 可能会有人认为这无关紧要. 如果不能够证明，我们可以通过简单改用另外一种符号记法来达到目的. 例如，可以使用有序对 $(\neg, \alpha)$ 和有序三元组 $(\alpha, \wedge, \beta)$ 之类的符号来取代通过连接的方式构造公式的方法. (事实上，这是一种非常精简的方法，但是不符合我们的习惯.) 这样，每个公式都有唯一的分解方法. 然而，下面我们要做的是证明没必要这样做.

引理 13A 每个合式公式具有同样多的左括号和右括号.

证明 在 1.1 节中，我们已经把它作为一个例子证明过了. ■

引理 13B 合式公式的任意一个真的初始段含有的左括号多于右括号. 这样，真的初始段不会成为一个合式公式.

证明 对具有该性质 (初始段的左括号多) 的合式公式的集合 S 使用归纳法. 只包括一个命题符号的合式公式没有符合条件的真的初始段，显然是成立的. 验证 S 在 $\mathcal{E}_\wedge$ 的作用下是封闭的，考虑 S 中的 α 与 β. $(\alpha \wedge \beta)$ 的真的初始段如下:

(1) (.
(2) $(\alpha_0$, 这里的 α_0 是 α 的真初始段.
(3) $(\alpha$.
(4) $(\alpha \wedge$.
(5) $(\alpha \wedge \beta_0$, 这里的 β_0 是 β 的真初始段.

1. 已经接受 $\bar{v}$ 的存在性的读者几乎可以忽略这一节，但最后一部分仍然是有用的，它是关于括号省略的问题.

(6) $(\alpha \wedge \beta$.

使用归纳假设，α、β 在 S 中 (第 2 种和第 5 种情况)，结论显然是正确的. 对其他几种形式的运算，讨论过程类似. ■

1.3.1 解析算法

对于给定的表达式，我们现在要描述下面的过程：(1) 确定该表达式是否是合法的合式公式；(2) 如果是，使用命题符号和公式构造运算建立它的生成树. 另外，还要验证这棵树是否是由这个合式公式*唯一确定的*. 这一点能够保证使用足够多的括号就能消除模糊性.

30

假定有一个表达式，我们要使用该表达式向下生成一棵树. 最初，树只有一个顶点，随着生成过程的进行，树开始从给定的表达式开始向下生长. 比如，在 1.1 节中的例子就是这样.

该算法包括以下 4 步：

(1) 当树的底部只有命题符号时，过程结束. (给定的表达式确实是合式公式，并且树已经生成.) 否则，选择其中一个非命题符号的表达式继续进行.

(2) 第 1 个符号必定 [1] 是 (. 第 2 个符号如果是否定符号则转到第 4 步，否则转到第 3 步.

(3) 从左边开始扫描表达式，直到遇到 $(\alpha$ 为止，这里的 α 是指一个非空的左右括号平衡的表达式 [2]. 则 α 是两个组成部分中的第一个，下一个符号必定是 $\wedge$、$\vee$、$\rightarrow$ 或者 $\leftrightarrow$ 中的一个，这是基本的联结符. 剩下的部分是 $\beta)$，一定 [1] 包括一个表达式 β 和一个右括号. 在当前树下添加两个新的顶点：α 是"左孩子"节点，β 是"右孩子"节点. 返回第 1 步.

(4) 表达式的前两个符号是 $(\neg$. 表达式的剩余部分是 $\beta)$，一定 [1] 包括一个表达式 β 和一个右括号). 在当前树下添加一个新的顶点：β 是孩子节点. 返回第 1 步.

以下是关于这个算法的正确性的讨论.

首先，对于任意给定的表达式，算法会在有限步内结束. 这是因为所有顶点对应的表达式都比其上一级的表达式短，因此树的深度可以由给定的表达式的长度限定.

其次，整个过程的选择都是唯一的. 比如，在第 3 步得到一个表达式 α. 我们不可能得到一个比 α 更短的表达式，原因在于比它短的式子不可能是左右括号平衡的 (引理 13A). 也不可能得到一个比 α 更长的表达式，那样的话就会得到一个括号平衡的初始段 (违背引理 13B). 因此，我们得到的肯定是 α. 同时，联结符号也是别无选择的. 这样算法构造的树就是*唯一*的.

31

再次，如果算法处理的是脚注所述的表达式，那么表达式就不是合式公式——算法拒绝处理这样的表达式. 因为在这种情况下，无法生成树.

最后，如果算法没有遇到脚注所述的表达式，那么给定的表达式确实是一个合法的合式公式. 这是由于可以根据算法得到该合式公式对应的树，由归纳可知，每个顶点都对应一个合式公式，根亦如此.

上述的第二条允许我们在形式语言中使用足够多的括号；每个合式公式都对应*唯一*的一棵树. 我们的形式语言具有"唯一可读性"，合式公式有唯一的树与之对应，而且我们也

1. 否则，表达式就不是合式公式，算法拒绝处理非合式公式的表达式.
2. 如果表达式结束时没有找到满足条件的 α，那么表达式不是合式公式，算法拒绝处理非合式公式的表达式.

知道构造该树的方法，只要有足够的空间，我们就可以执行这个算法.

现在我们回到真值指派 v 的扩展 $\bar{v}$ 的存在性问题上来：树的唯一性是这个问题的关键. 对于任意的合式公式 φ，只能构造出唯一的一棵树. 给树中的每个顶点按照 $\bar{v}(\alpha)$ 进行真值指派，可以明确地得到一个 $\bar{v}(\varphi)$ 的值. 该算法的功能满足 1.2 节中的条件 0~5. 不仅如此，对于给定的 φ 及 v 在命题符号中的赋值，我们也就知道计算 $\bar{v}(\varphi)$ 的方法.

按照定理 12A 中所述，我们可以使用解析算法建立一个函数 $\bar{v}$. 并且仅有一个这样的函数 $\bar{v}$，可与 1.2 节的习题 14 进行比较.

$\bar{v}$ 的存在性成为我们的焦点的原因是，在 1.2 节中 $\bar{v}$ 是用递归的方式描述的. 即 $\bar{v}(\varphi)$ 是通过函数 $\bar{v}$ 本身对较小的公式的赋值计算得到的. 在下一节中，我们将使用递归的方法更一般地定义函数. 通过更抽象的处理，可以更好地减少风险.

1.3.2 波兰记法

不使用括号，也可以避免模糊性. 这可以通过一个非常简单的方法实现. 比如，可以用 $\wedge\alpha\beta$ 代替 $(\alpha\wedge\beta)$. 设 P 合式公式的集合是由命题符号和 5 种运算得到的：

$$\mathcal{D}_{\neg}(\alpha)=\neg\alpha,\qquad \mathcal{D}_{\vee}(\alpha,\beta)=\vee\alpha\beta,$$
$$\mathcal{D}_{\wedge}(\alpha,\beta)=\wedge\alpha\beta,\qquad \mathcal{D}_{\rightarrow}(\alpha,\beta)=\rightarrow\alpha\beta,$$
$$\mathcal{D}_{\leftrightarrow}(\alpha,\beta)=\leftrightarrow\alpha\beta.$$

32

例如，一个 P 合式公式为

$$\rightarrow\ \wedge\ \mathbf{AD}\vee\ \neg\mathbf{B}\leftrightarrow\mathbf{CB}.$$

很明显，我们需要一个算法来分析公式的结构. 即使是很短的公式，也需要知道它是如何构造出来的. 2.3 节给出了针对这类表达式的唯一可读性定理.

波兰的逻辑学家 Łukasiewicz 首先使用这种书写公式的方法 (不过他不是使用 $\neg$、$\wedge$、$\vee$、$\rightarrow$ 和 $\leftrightarrow$ 这些符号，而是使用 N、K、A、C 和 E)，这种记法适于自动处理过程. 计算机编译器往往首先要将公式转化为波兰记法再进行处理.

1.3.3 省略括号

今后，在书写合式公式时，不必写出每一个括号. 为了使公式更加简洁紧凑，我们采用以下约定：

(1) 最外层的括号可以省略. 比如，用 $\mathbf{A}\wedge\mathbf{B}$ 表示 $(\mathbf{A}\wedge\mathbf{B})$.

(2) 否定符号外面的括号可以尽量省略. 例如，$\neg\mathbf{A}\wedge\mathbf{B}$ 就是 $(\neg\mathbf{A})\wedge\mathbf{B}$，即 $((\neg\mathbf{A})\wedge\mathbf{B})$，但不是 $(\neg(\mathbf{A}\wedge\mathbf{B}))$.

(3) 应用析取与合取符号时，尽可能保持公式的简短. 例如，

$$\mathbf{A}\wedge\mathbf{B}\rightarrow\neg\mathbf{C}\vee\mathbf{D}\quad 即\quad ((\mathbf{A}\wedge\mathbf{B})\rightarrow((\neg\mathbf{C})\vee\mathbf{D})).$$

(4) 重复使用同一种连接符号时，先计算右边.

$$\alpha\wedge\beta\wedge\gamma\quad 即\quad \alpha\wedge(\beta\wedge\gamma),$$
$$\alpha\rightarrow\beta\rightarrow\gamma\quad 即\quad \alpha\rightarrow(\beta\rightarrow\gamma).$$

必须承认的是这些约定违背了前面我们说的表达式的形成规则. 我们可以不再使用那些形成规则, 因为我们对非合式公式的形成没有任何兴趣.

习题

1. 使用本节最后的约定重写 1.2 节结尾处列出的重言式, 减少公式中的括号数量.
2. [33] 给出合式公式 α 和 β 及表达式 γ 与 δ 的例子, 使得 $(\alpha \wedge \beta) = (\gamma \wedge \delta)$ 但 $\alpha \neq \gamma$.
3. 证明引理 13B 中关于运算 $\mathcal{E}_{\neg}$ 的情形.
4. 假定将合式公式的定义修改为去掉其中所有的*右*括号. 对于

$$((\mathbf{A} \wedge (\neg\, \mathbf{B})) \to (\mathbf{C} \vee \mathbf{D}))$$

   我们写成

$$((\mathbf{A} \wedge (\neg\, \mathbf{B} \to (\mathbf{C} \vee \mathbf{D}.$$

   证明这种记法具有唯一可读性 (即每个合式公式只有一个可能的分解方法). *提示*: 这种表达式具有的括号数和联结符号数相同.
5. 汉语中经常使用*二部连接词*(two-part connective): "……和……", "要么……要么……", "如果……那么……", 这种表达方法对汉语的唯一可读性有什么影响?
6. 我们已经给出一种自上而下生成树的算法来分析合式公式. 当然这样的树也可以用自下而上的方法来构造. 这要通过扫描公式的最内层的括号对来实现. 请给出这种算法的一个完整描述.
7. 假定左右括号是无法区分的, 如用 $|\alpha \vee |\beta \wedge \gamma||$ 取代 $(\alpha \vee (\beta \wedge \gamma))$. 公式是否还具有唯一可读性?

1.4　归纳与递归[1]

1.4.1　归纳

归纳是一类经常出现在逻辑和数学其他分支中的特殊的构造方法. 通过对集合 U 的某些初始元素重复使用几种运算, 可以构造 U 的一个特定的子集. 这个子集是包含初始元素的最小集合, 并且它对相应的运算是封闭的. 其中的元素可以通过对 U 中的初始元素有限次地使用相关运算而得到. [34]

在我们感兴趣的特例中, U 是一个表达式的集合, 它的初始元素是命题符号, 相关运算是 $\mathcal{E}_{\neg}$, $\mathcal{E}_{\wedge}$, 等. 这样构造出来的就是合式公式的集合. 后面, 我们将会碰到其他的一些特殊例子, 会有助于我们更抽象地理解目前的特例.

为了简化讨论过程, 我们考虑初始集合 $B \subseteq U$ 和只包含两个函数 f 与 g 的函数类 $\mathcal{F}$, 其中:

$$f : U \times U \to U, \quad g : U \to U.$$

即 f 是 U 上的二元运算, 而 g 是一元运算. (事实上, $\mathcal{F}$ 可以是无限的. 这里我们讨论的简化情形可以应用到更为广泛的情形中. $\mathcal{F}$ 可以是 U 上任意关系的集合, 第 2 章中用得会更多一些. 这里的例子是为了易于理解, 并且足以表达一般的思想就够了. 习题 3 给出了更一般的情形.)

---

1. 本节中涉及的概念是非常重要的, 经常应用在数学的各个领域中. 读者可以选择稍后再阅读本节, 但是最好不要跳过本节.

若集合 B 包括元素 a 和 b, 那么在我们想构造的集合 C 中要有以下元素 (当然, 不止这些),

$$b,\quad f(b,b),\quad g(a),\quad f(g(a),\quad f(b,b)),\quad g(f(g(a),f(b,b))).$$

当然, 这些元素可能会有重复. 其基本思想可以这样类比: 有了砖头和泥灰, 集合 C 就是使用这些原料做出来的一切东西.

对 C 的规范定义有两个. "自上向下" 的定义如下: 我们称 U 的子集 S 在 f 和 g 的作用下是*封闭的*当且仅当只要 x 与 y 属于 S, 那么 $f(x,y)$ 和 $g(x)$ 也属于 S. 称 S 是*归纳的*当且仅当 $B\subseteq S$ 并且 S 在 f 和 g 的作用下是封闭的. 令 C^* 是 U 的所有归纳子集的交集, 则 $x\in C^*$ 当且仅当 x 属于 U 的每个归纳子集. 不难看出 C^* 本身也是归纳的. 另外, 由于 C^* 包含在其他任意一个归纳子集中, 所以它是最小的.

第二个 (也是等价的) 定义是"自下而上"给出的. 设 C_* 是由 B 中的元素有限次使用 f 和 g 得到的所有元素的集合. 临时定义一个*构造序列*是 U 中元素的有限序列 $\langle x_1,x_2,\cdots,x_n\rangle$, 使得对每个 $i\leqslant n$, 它至少满足以下三条之一:

$$x_i\in B,$$
$$x_i=f(x_j,x_k)\quad 对于某些\ j<i,k<i,$$
$$x_i=g(x_j)\quad 对于某些\ j<i.$$

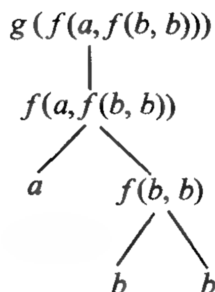
换句话说, 序列中的每个元素或者在 B 中, 或者是 f 和 g 对序列中出现在该元素*前*的某些元素作用的结果. 设 x 是某个构造序列中的最后一个元素, C_* 是所有这样的 x 组成的集合.

35

设 C_n 是构造序列长度为 n 的最后一个元素是 x 的集合, 那么 $C_1=B$,

$$C_1\subseteq C_2\subseteq C_3\subseteq\cdots,$$

并且 $C_*=\bigcup_n C_n$. 例如, $g(f(a,f(b,b)))$ 在 C_5 中, 因而也在 C_* 中, 这可以通过构造下面的树来证明:



将这棵树压缩成线性序列, 我们就可以得到 $g(f(a,f(b,b)))$ 的构造序列.

例 (1) 自然数. 设 U 是实数集, $B=\{0\}$, 运算 S 定义为 $S(x)=x+1$, 则

$$C_*=\{0,1,2,\cdots\}.$$

C_* 是自然数集, 恰好是从 0 开始反复使用后继运算得到的数的集合.

(2) 整数. 设 U 是实数集, $B=\{0\}$, 这次使用两种运算: 后继运算 $S(x)=x+1$ 和前驱运算 $P(x)=x-1$. 则 C_* 包括所有的整数,

$$C_*=\{\cdots,-2,-1,0,1,2,\cdots\}.$$

注意: C_* 的元素 2 有不止一种生成方式, 如 $S(S(0))$ 和 $S(P(S(S(0))))$.

(3) 代数函数. 设 U 是所有的定义域与值域都是实数集的函数的集合, B 包括恒等函数和所有常值函数. 令 $\mathcal{F}$ 包含 (函数的) 加、减、乘、除和求根运算. 则 C_* 是代数函数类.

(4) 合式公式. 设 U 是所有表达式的集合, B 是命题符号的集合. $\mathcal{F}$ 含有表达式上的 5 种公式构造运算: $\mathcal{E}_\neg$, $\mathcal{E}_\wedge$, $\mathcal{E}_\vee$, $\mathcal{E}_\rightarrow$, $\mathcal{E}_\leftrightarrow$. 则 C_* 是合式公式的集合. [36]

下面我们验证前面的两个定义实际上是等价的, 即 $C^*=C_*$.

首先验证 $C^*\subseteq C_*$. 我们只需证明 C_* 是归纳的, 即 $B\subseteq C_*$ 并且 C_* 在函数作用下是封闭的. 显然 $B=C_1\subseteq C_*$. 如果 x 和 y 都在 C_* 中, 那么将其构造序列进行连接, 并添加一个新的元素 $f(x,y)$, 这样我们就得到一个新的构造序列, 并将 $f(x,y)$ 加在了 C_* 中. 类似地, C_* 在 g 的作用下也是封闭的.

最后, 还要证明 $C_*\subseteq C^*$, 这就要考察 C_* 的一个元素和它的构造序列 $\langle x_0,\cdots,x_n\rangle$. 对 i 进行归纳, 可以得到 $x_i\in C^*$ $(i\leqslant n)$. 首先, $x_0\in B\subseteq C^*$. 至于归纳步骤我们可以使用 C^* 在运算作用下封闭的特性. 这样, 就得到

$$\bigcup_n C_n=C_*=C^*=\bigcap\{S|S\text{是归纳的}\}.$$

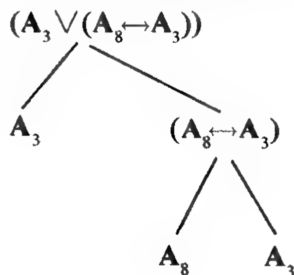
(附加说明: 公理集合论中自然数通常是自上而下定义的. 如果把我们现在学习的内容嵌入到公理集合论中, 那么 C_*(利用有限性得到自然数的定义) 的定义与 C^* 的定义就不会有真正的区别了. 可是, 我们现在不是在学习公理集合论, 而是非形式化的数学. 并且自然数的概念也是可靠的、直观易懂的.)

由于 $C^*=C_*$, 我们可以简单地使用 C 来表示由 B 中的元素通过 $\mathcal{F}$ 中的运算得到的集合. 今后, 经常会使用以下原理来证明结论.

归纳法则 假设 C 是由 B 中的元素通过 $\mathcal{F}$ 中的函数生成的, 若 S 是 C 的子集, S 包含 B 并且在 $\mathcal{F}$ 中的运算作用下是封闭的, 那么 $S=C$.

证明 S 是归纳的, 因此 $C=C^*\subseteq S$. 易得. ■

当然, 上面的例 4 是我们感兴趣的特例. C 是由命题符号和公式构造运算生成的合式公式的类, 这个例子具有有趣的特点. α 与 β 都是 $\mathcal{E}_\wedge(\alpha,\beta)$, 即, $\alpha\wedge\beta$ 的真子段, 更一般地, 从合式公式的生成树可以看出每个组成部分都是最终结果的真子段. [37]



例如，对于仅用到 $\{\mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_5\}$ 中的命题符号和 $\{\neg, \rightarrow\}$ 中的联结符的表达式，我们暂且称之为特定表达式，那么非特定的表达式可能会用到 $\mathbf{A}_9$ 和 $\mathcal{E}_\wedge$. 事实上，每个特定的合式公式都属于由 $\{\mathbf{A}_2, \mathbf{A}_3, \mathbf{A}_5\}$ 通过运算 $\mathcal{E}_\neg, \mathcal{E}_\rightarrow$ 生成的集合 C_S 中. (可以使用归纳法则证明每个合式公式或者属于 C_S 或者不是特定的.)

1.4.2 递归

现在来考虑更抽象的情形. 设有集合 U(比如是所有表达式的集合), U 的子集 B(比如是命题符号的集合), 和两个函数 f 和 g, 其中

$$f : U \times U \rightarrow U,\ g : U \rightarrow U.$$

C 是由 B 在函数 f, g 作用下生成的集合.

现在我们来递归地定义一个 C 上的函数. 假定有

(1) 对于 $x \in B$, 计算 $\bar{h}(x)$ 的规则.

(2a) 利用 $\bar{h}(x)$ 和 $\bar{h}(y)$ 计算 $\bar{h}(f(x, y))$ 的规则.

(2b) 利用 $\bar{h}(x)$ 计算 $\bar{h}(g(x))$ 的规则.

(如 1.2 节中讨论的情形，其中 $\bar{h}$ 是 B 的一个真值指派的扩展.) 不难看出，最多有一个 C 上的函数 $\bar{h}$ 满足所有给定的要求.

然而，也有因为规则的矛盾而导致这样的函数 $\bar{h}$ 不存在的情形. 例如，设

$$\begin{aligned} U &= \text{实数集} \\ B &= \{0\} \\ f(x, y) &= x \cdot y \\ g(x) &= x + 1. \end{aligned}$$

那么 C 就是自然数集. 假如对 $\bar{h}$ 加上以下要求：

(1) $\bar{h}(0) = 0$.

(2a) $\bar{h}(f(x, y)) = f(\bar{h}(x), \bar{h}(y))$.

(2b) $\bar{h}(g(x)) = \bar{h}(x) + 2$.

那么满足条件的函数 $\bar{h}$ 就不存在了. (试计算 $\bar{h}(1)$, 请注意同时有 $1 = g(0)$ 和 $1 = f(g(0), g(0))$.)

代数中也会有类似的情形 [1]. 设群 G 是由 B 和群的乘法运算及逆运算生成的，那么从 B 到群 H 的任意映射不一定能够扩展成从整个群 G 到 H 的同态. 但是，如果 G 恰好是以 B 为生成集的自由群，那么这个映射就可以扩展为群同态. 38

称 C 是由 B 在 f 和 g 的作用下自由生成的，当且仅当除了满足生成的要求外，f 和 g 在 C 上的限制 f_C 和 g_C 必须满足以下条件：

(1) f_C 和 g_C 是一对一的.

(2) f_C 的值域，g_C 的值域和集合 B 是两两不交的.

---

1. 我们希望这些例子对于有代数基础的读者有用. 其他读者了解这些例子仅用于说明问题，不是本课程的核心内容即可.

本节的主要结论是递归定理，它说明了如果 C 是自由生成的，那么 B 上的函数 h 总是可以扩展到 C 上的函数 $\bar{h}$，且满足上述规则.

递归定理 设 U 的子集 C 是由 B 在 f 和 g 的作用下自由生成的，其中

$$\begin{aligned} f &: U \times U \to U, \\ g &: U \to U. \end{aligned}$$

设 V 是集合，函数 F、G 和 h 满足

$$\begin{aligned} h &: B \to V, \\ F &: V \times V \to V, \\ G &: V \to V. \end{aligned}$$

那么，存在唯一的函数

$$\bar{h} : C \to V$$

使得

(i) 对 B 中的 x，$\bar{h}(x) = h(x)$；

(ii) 对 C 中的 x，y，

$$\begin{aligned} &\bar{h}(f(x,y)) = F(\bar{h}(x), \bar{h}(y)), \\ &\bar{h}(g(x)) = G(\bar{h}(x)). \end{aligned}$$

从代数的角度看，该定理的结论说明了从 B 到 V 中的任何映射 h 都可以扩展到从 C (带有运算 f 和 g) 到 V(带有运算 F 和 G) 的同态.

如果递归定理的内容不是显而易见的，那么我们可以试着通过涂色的方式来理解，希望函数 $\bar{h}$ 给 C 中每个元素涂上某种颜色. 首先，我们已知

(1) h，给定 B 的初始元素的涂色方案.

39

(2) F，根据 x 与 y 的颜色得到 $f(x,y)$ 的颜色 (即给出关于 $\bar{h}(x)$，$\bar{h}(y)$ 的 $\bar{h}(f(x,y))$).

(3) G，类似地给出从 x 的颜色得到 $g(x)$ 的颜色的方法.

问题是这可能会产生冲突，比如，F 要给某个点涂绿色，而 G 却要给该点涂红色 (极有可能对某些 $x,y,z,f(x,y)$ 会和 $g(z)$ 相等). 可是，如果 C 是自由生成的，这个问题就不存在了.

例 重新考虑前一节中的例子.

(1) 如果 $B = \{0\}$ 只含一个运算，即后继运算 S，那么 C 就是自然数集 $\mathbb{N}$. 由于后继运算是一对一的并且 0 不在其值域中，所以 C 是由 $\{0\}$ 通过后继运算自由生成的. 因此，通过递归定理，对于任意集合 V，任意的 $a \in V$ 和任意的 $F : V \to V$，存在唯一的 $\bar{h} : \mathbb{N} \to V$ 满足 $\bar{h}(0) = a$ ，并且对每个 $x \in \mathbb{N}$，$\bar{h}(S(x)) = F(\bar{h}(x))$. 例如，存在唯一的 $\bar{h} : \mathbb{N} \to \mathbb{N}$ 满足 $\bar{h}(0) = 0$，且 $\bar{h}(S(x)) = 1 - \bar{h}(x)$. 该函数在偶数上的取值为 0，而在奇数上的取值为 1.

(2) 整数是由 $\{0\}$ 通过后继运算和前驱运算的作用生成的，但不是自由生成的.

(3) 代数函数同样不是自由生成的.

(4) 合式公式是由命题符号通过 5 种公式构造运算生成的. 这一点在前一节的算法中不是显而易见的, 现在来看一下.

唯一可读性定理 5 个合式公式集合上的公式构造运算, 具有以下性质:

(a) 具有两两不相交的值域, 并且和命题符号集也是不交的;

(b) 是一对一的.

换句话说, 合式公式的集合是由命题符号集在 5 种运算的作用下 *自由* 生成的.

证明 为了证明对 $\mathcal{E}_\wedge$ 的限制是一对一的, 我们假定

$$(\alpha \wedge \beta) = (\gamma \wedge \delta),$$

其中 α, β, γ 和 δ 是合式公式, 删除上式中左右两边的第一个字符, 得到

$$\alpha \wedge \beta) = \gamma \wedge \delta).$$

这样我们就有 $\alpha = \gamma$, 否则其中一个就是另一个的真的初始段 (违背引理 13B). 接着, 我们就有 $\beta = \delta$. 对于 $\mathcal{E}_\vee$, $\mathcal{E}_\rightarrow$, $\mathcal{E}_\leftrightarrow$, 可以类似地进行讨论, 而对 $\mathcal{E}_\neg$ 的讨论则是简单的. 40

类似的推理可以得到运算值域的互不相交性. 例如, 如果

$$(\alpha \wedge \beta) = (\gamma \rightarrow \delta)$$

其中 α、β、γ 和 δ 是合式公式, 如前所述, 可以得到 $\alpha = \gamma$. 但是, 这就意味着 $\wedge = \rightarrow$, 这显然是矛盾的. 因此 $\mathcal{E}_\wedge$ 和 $\mathcal{E}_\rightarrow$ 具有不同的值域. 类似地, 可以讨论其他的二元运算联结符.

其余的情形就简单多了. 如果 $(\neg\, \alpha) = (\beta \wedge \gamma)$, 那么 β 就要以 $\neg$ 为开始, 这样就不是合式公式了. 因为命题符号不能是以 (开头的符号序列. ■

现在, 回到扩展真值指派 v 到 $\bar{v}$ 的问题上来. 首先考虑一种特殊情形, 即 v 是所有命题符号的真值指派. 根据唯一可读性定理和递归定理, 一定存在对所有合式公式集合的唯一扩展 $\bar{v}$, 它具有我们所希望的性质.

接下来, 我们考虑一般的情形: v 是命题符号集合 S 的真值指派. 由唯一可读性定理可知, 集合 $\bar{S}$ 是 S 在 5 种公式构造运算作用下生成的集合. 这样根据递归定理, 真值指派 v 存在唯一的扩展 $\bar{v}$, 满足所需的条件.

例 我们可以运用递归定理来证明, 存在唯一的函数 $\bar{h}$, 它的定义域是合式公式集合, 并且满足:

$$\begin{aligned} \bar{h}(A) &= 1 \quad \text{对于命题符号}A, \\ \bar{h}((\neg\, \alpha)) &= 3 + \bar{h}(\alpha), \\ \bar{h}((\alpha \wedge \beta)) &= 3 + \bar{h}(\alpha) + \bar{h}(\beta). \end{aligned}$$

并且对 $\vee$, $\rightarrow$, $\leftrightarrow$ 也类似. 这个函数可以用于求每个合式公式的长度.

递归定理的证明 我们的思路是令 $\bar{h}$ 为多个逼近函数的并. 如果一个函数 v (将 C 中的一部分映射到 V 中) 满足 $\bar{h}$ 上的条件 (i) 和 (ii), 我们就称 v 是可以接受的. 准确地说, v 是可接受的当且仅当 v 的定义域是 C 的子集, 值域是 V 的子集, 并且对于 C 中的 x 和 y 有:

(i′) 如果 x 属于 B 且属于 v 的定义域，那么 $v(x) = h(x)$；

(ii′) 如果 $f(x,y)$ 属于 v 的定义域，那么 x，y 也属于 v 的定义域并且 $v(f(x,y)) = F(v(x),v(y))$. 如果 $g(x)$ 属于 v 的值域，那么 x 也属于 v 的值域，并且 $v(g(x)) = G(v(x))$.

41

设 K 是所有可接受的函数的集合，令 $\bar{h} = \bigcup K$，那么，

$$\begin{aligned} \langle x,z\rangle \in \bar{h} \quad & \text{iff} \quad \langle x,z\rangle \text{ 属于某个可接受的} v, \\ & \text{iff} \quad \text{对某个可接受的} v, v(x) = z. \qquad (*) \end{aligned}$$

我们说 $\bar{h}$ 满足我们的要求，其证明与集合论有关，共包括 4 步. 首先，我们给出这 4 步的大纲.

(1) 证明 $\bar{h}$ 是一个函数 (即是单值).

设 $S = \{x \in C|$ 最多存在一个 z，使得 $\langle x,z\rangle \in \bar{h}\}$

$= \{x \in C|$ 所有定义在 x 上的可接受函数在 x 点的取值相同 $\}$

使用 (i′) 和 (ii′)，容易证明 S 是归纳的. 因此，$S = C$，并且 $\bar{h}$ 是一个函数.

(2) 证明 $\bar{h} \in K$；也就是说，$\bar{h}$ 本身是一个可接受的函数. 根据 $\bar{h}$ 的定义，以及 $\bar{h}$ 也是一个函数的事实，容易证明这一点.

(3) 证明 $\bar{h}$ 是定义在整个 C 上的. 这只需要证明 $\bar{h}$ 的定义域是归纳的. 这里用到了子集 C 是自由生成的假设. 比如，有一种情况是这样的：假定 x 在 $\bar{h}$ 的定义域中，那么 $\bar{h};\langle g(x), G(\bar{h}(x))\rangle$ 是可接受的. (在证明其可接受的时候用到自由生成.) 因此，$g(x)$ 也在 $\bar{h}$ 的定义域中.

(4) 证明 $\bar{h}$ 是唯一的. 对于给定的两个这样的函数，设它们在集合 S 上取值相等，那么 S 是归纳的，因此等于 C. ■

下面是详细证明.

(1) 如上所述，设

$S = \{x \in C|$ 最多存在一个 z，使得 $\langle x,z\rangle \in \bar{h}\}$

$= \{x \in C|$ 所有定义在 x 上的可接受函数在 x 点的取值相同 $\}$

为了证明 S 是归纳的，首先考虑 B 中的某个 x，假设 v_1 和 v_2 是在 x 上定义的可接受函数，我们要证明 $v_1(x) = v_2(x)$. 根据条件 (i′)，$v_1(x)$ 与 $v_2(x)$ 必须等于 $h(x)$，因此，$v_1(x) = v_2(x)$. 这说明 $x \in S$，还因为 x 是 B 的任意元素，所以有 $B \subseteq S$.

其次，验证 S 在 f 和 g 的作用下是封闭的. 设 x 和 y 是 S 中的元素，我们要判断的是 $f(x,y)$ 是否在 S 中. 假定 v_1 和 v_2 是定义在 $f(x,y)$ 上的可接受函数，也要证明这两个函数在这一点上取值相等. 然而，条件 (ii′) 表明 $v_1(f(x,y)) = F(v_1(x),v_1(y))$，且 $v_2(f(x,y)) = F(v_2(x),v_2(y))$. 由于 x 和 y 在 S 中，因此有 $v_1(x) = v_2(x)$ 和 $v_1(y) = v_2(y)$. 这样就得到 $v_1(f(x,y)) = v_2(f(x,y))$. 这就证明了 $f(x,y) \in S$，因此 S 在 f 下是封闭的. 类似地，可以证明 S 在 g 的作用下也是封闭的.

42

因而 S 是归纳的，且 $S = C$. 这就证明了 $\bar{h}$ 是单值的，即是一个函数. 又因为 $\bar{h}$ 包含了每个可以接受的函数作为其子集，因此我们说：

$$\bar{h}(x) = v(x),$$

其中 v 是可接受的函数且 $x \in \operatorname{dom} v$.

(2) 证明 $\bar{h}$ 是可接受的. 显然 $\mathrm{dom}\ \bar{h} \subseteq C$ 且 $\mathrm{ran}\ \bar{h} \subseteq V$(由 (*)), 我们仅仅证明了 $\bar{h}$ 是一个函数, 还需要证明它满足条件 (i′) 和 (ii′).

首先, 验证 (i′). 设 $x \in B$ 且 $x \in \mathrm{dom}\ \bar{h}$(因此, $\langle x, \bar{h}(x)\rangle \in \bar{h}$), 那么必定存在某个可接受函数 v 使得 $v(x) = \bar{h}(x)$. 由于 v 满足条件 (i′), 故我们有 $v(x) = h(x)$, 进而 $\bar{h}(x) = h(x)$. 因此, $\bar{h}$ 满足条件 (i′).

其次, 验证 (ii′). 设 $f(x,y) \in \mathrm{dom}\ \bar{h}$, 必定存在某个可接受函数 v 使得 $v(f(x,y)) = \bar{h}(f(x,y))$. 由于 v 满足条件 (ii′), 故我们有 $v(f(x,y)) = F(v(x), v(y))$. 由于 $\bar{h}(x) = v(x)$ 且 $\bar{h}(y) = v(y)$, 因此, $\bar{h}(f(x,y)) = v(f(x,y)) = F(v(x), v(y)) = F(\bar{h}(x), \bar{h}(y))$. 采用同样的方法我们可以得到, 只要 $g(x) \in \mathrm{dom}\ \bar{h}$, 就有 $\bar{h}(g(x)) = G(\bar{h}(x))$. 这样, $\bar{h}$ 满足条件 (ii′), 因而也是可接受的.

(3) 证明 $\bar{h}$ 的定义域是归纳的. 首先, 考虑 B 中的点 x, 我们说集合 $\{\langle x, h(x)\rangle\}$ 是一个 (小的) 可接受函数. 因为它显然满足条件 (i′); 同时它也满足条件 (ii′), 这是由于 $x \notin \mathrm{ran}\ f_C$ 并且 $x \notin \mathrm{ran}\ g_C$. 这样, $\{\langle x, h(x)\rangle\}$ 是可接受的, 进而包含在 $\bar{h}$ 中. 因此 $x \in \mathrm{dom}\ \bar{h}$, $B \subseteq \mathrm{dom}\ \bar{h}$.

进一步证明 $\mathrm{dom}\ \bar{h}$ 在 f 和 g 的作用下是封闭的. 为此我们考虑 $\mathrm{dom}\ \bar{h}$ 中的任意元素 s 和 t, 我们希望 $f(s,t) \in \mathrm{dom}\ \bar{h}$ 能够成立. 否则, 令

$$v = \bar{h} \cup \{\langle f(s,t), F(\bar{h}(s), \bar{h}(t))\rangle\},$$

即 v 是在 $\bar{h}$ 后添加这个附加的有序对后的结果. 显然, v 是函数, $\mathrm{dom}\ v \subseteq C$, 且 $\mathrm{ran}\ v \subseteq V$. 我们说 v 满足条件 (i′) 和 (ii′).

43

这是因为: 对于 (i′), 如果 $x \in B \cap \mathrm{dom}\ \bar{h}$, 由自由生成的假设可知 $x \neq f(s,t)$, 因此, $x \in \mathrm{dom}\ \bar{h}$ 并且 $v(x) = \bar{h}(x) = h(x)$.

对于 (ii′), 假定对于 C 中的某个 x, y, 有 $f(x,y) \in \mathrm{dom}\ v$. 如果 $f(x,y) \in \mathrm{dom}\ \bar{h}$, 由于 $\bar{h}$ 是可接受的, 则 $v(f(x,y)) = \bar{h}(f(x,y)) = F(\bar{h}(x), \bar{h}(y)) = F(v(x), v(y))$. 还有一种可能就是 $f(x,y) = f(s,t)$, 由自由性可知, $x = s$, $y = t$, 而这些点都是在 $\mathrm{dom}\ \bar{h} \subseteq \mathrm{dom}\ v$ 中的. 由构造过程可知:

$$\begin{aligned} v(f(s,t)) &= F(\bar{h}(s), \bar{h}(t)) \\ &= F(v(s), v(t)). \end{aligned}$$

最后, 假设对于 C 中的 x, 有 $g(x) \in \mathrm{dom}\ v$. 那么根据自由性, 有 $g(x) \neq f(s,t)$. 因此 $g(x) \in \mathrm{dom}\ \bar{h}$, $v(g(x)) = \bar{h}(g(x)) = G(\bar{h}(x)) = G(v(x))$.

这样就得到 v 是可接受的, 然而这意味着 $v \subseteq \bar{h}$, 因而 $f(s,t) \in \mathrm{dom}\ \bar{h}$.

类似地, 可以证明 $\mathrm{dom}\ \bar{h}$ 在 g 作用下也是封闭的. 进而 $\mathrm{dom}\ \bar{h}$ 是归纳的, 并且与 C 相等.

(4) 证明 $\bar{h}$ 是唯一的. 假定两个函数 $\bar{h}_1$ 和 $\bar{h}_2$ 都满足定理结论, 设它们在集合 S 上取值相等, 即

$$S = \{x \in C | \bar{h}_1(x) = \bar{h}_2(x)\}.$$

那么不难验证 S 是归纳的, 因此 $S = C$, $\bar{h}_1 = \bar{h}_2$. ■

最后，我们对归纳和递归作个总结：本节中的归纳法则不是唯一的．我们完全有可能通过对表达式的长度，表达式中联结符号出现的次数等进行归纳 (及递归定义)，也会有相应的证明．这样的方法从本质上说并非基本的，但在某些情况下却是必要的．

习题

1. 设 C 是由集合 $B=\{a,b\}$ 在二元运算 f 和一元运算 g 的作用下生成的．试列出 C_2 中的所有元素．C_3 和 C_4 中各有多少元素？
2. 证明 $(\mathbf{A}_3 \to \wedge\mathbf{A}_4)$ 不是合式公式．
3. 本节中关于要求 $\mathcal{F}$ 仅是 U 上的关系类的讨论可以进行推广．C_* 如前定义，但是如果对于每个 $i \leqslant n$，我们有 $x_i \in B$ 或者对某个 $R \in \mathcal{F}$ 以及某些小于 i 的数 $j_1,\cdots,j_k$ 有 $\langle x_{j_1},\cdots,x_{j_k},x_i\rangle \in R$，那么我们说 $\langle x_0,x_1,\cdots,x_n\rangle$ 是一个构造序列．请给出 C^* 的正确定义，并证明 $C^*=C_*$．

44

1.5　命题联结词

我们已经对 5 种命题联结符号做了较为深入的学习．即使没有对联结词给出一般的定义，我们也能够想到不可能只有这 5 种联结词．问题是对形式语言添加更多的联结词会有多大的益处？从这 5 种联结词中去掉某一个，其损失又是什么？

本节的目的就是要明确这些问题，并给出答案．首先，考虑一个不规范的例子：向现有的形式语言中添加一个新的三元命题联结符 #，称之为多数决定符号．只要 α，β，γ 是合式公式，那么表达式 $(\#\alpha\beta\gamma)$ 就是合式公式．换句话说，现在有了第 6 个公式构造运算：

$$\mathcal{E}_\#(\alpha,\beta,\gamma)=(\#\alpha\beta\gamma).$$

接下来我们给出该符号的解释，即给定了 $\bar{v}(\alpha)$，$\bar{v}(\beta)$ 和 $\bar{v}(\gamma)$ 的值，如何计算 $\bar{v}\left((\#\alpha\beta\gamma)\right)$．我们给出如下定义：

$$\bar{v}((\#\alpha\beta\gamma)) \text{ 与 } \bar{v}(\alpha),\ \bar{v}(\beta) \text{ 和 } \bar{v}(\gamma) \text{ 中多数的取值相同.}$$

从如下严格意义上说，这样的扩展并不真会带来什么好处，对于扩展后的语言中的任意一个合式公式，原来语言中一定有一个合式公式与之重言等价．(当然，原语言中的合式公式比扩展语言中的合式公式可能要长得多．) 下面会证明这一点 (更一般的情形)；这里只需要注意 $(\#\alpha\beta\gamma)$ 实际上重言等价于

$$(\alpha\wedge\beta)\vee(\alpha\wedge\gamma)\vee(\beta\wedge\gamma).$$

(附加说明：我们断定从 $\langle\bar{v}(\alpha),\bar{v}(\beta),\bar{v}(\gamma)\rangle$ 可以计算出 $\bar{v}((\#\alpha\beta\gamma))$，这是起到决定性作用的一点．在日常用语中，经常用到类似“……是可能的”或者“我相信……”这样的一元运算．如果将这种运算中的某一个作用于一个句子以产生一个新的句子，则新的句子的正误就不能仅由原来句子的正误来决定．)

在推广上面这个例子时，形式语言不仅没有帮助反倒是个麻烦．我们可以只用函数来重新描述这个问题．*k* 元布尔函数 是指从 $\{F,T\}^k$ 到 $\{F,T\}$ 的函数．(所谓的布尔函数就是对某个 k 而言的 *k* 元布尔函数，我们可以认为 F 和 T 本身就是 0 元布尔函数．) 有些布尔函数可以使用方程进行定义 (其中 $X\in\{F,T\}$)：

45

$$
\begin{gathered}
I_i^n(X_1,\cdots,X_n)=X_i,\\
N(F)=T,\qquad N(T)=F,\\
K(T,T)=T,\qquad K(F,X)=K(X,F)=F,\\
A(F,F)=F,\qquad A(T,X)=A(X,T)=T,\\
C(T,F)=F,\qquad C(F,X)=C(X,T)=T,\\
E(X,X)=T,\qquad E(T,F)=E(F,T)=F.
\end{gathered}
$$

从合式公式 α 也可以产生布尔函数. 例如, 如果 α 是合式公式 $\mathbf{A}_1\wedge\mathbf{A}_2$, 那么我们可以给出表 1-5. 表中的 2^2 行对应于 $\{\mathbf{A}_1,\mathbf{A}_2\}$ 的 2^2 个真值指派 $\overrightarrow{X}$. 对于 2^2 个 $\overrightarrow{X}$ 中的每一个而言, α 中命题符号的取值由 $\overrightarrow{X}$ 给定时 α 的值也就确定了, 而 $B_\alpha(\overrightarrow{X})$ 的值就取 α 的值.

表 1-5

$\mathbf{A}_1$	$\mathbf{A}_2$	$\mathbf{A}_1\wedge\mathbf{A}_2$	
F	F	F	$B_\alpha(F,F)=F$
F	T	F	$B_\alpha(F,T)=F$
T	F	F	$B_\alpha(T,F)=F$
T	T	T	$B_\alpha(T,T)=T$

通常我们说 α 是合式公式时, 它所含有的命题符号最多是 $\mathbf{A}_1,\cdots,\mathbf{A}_n$. n 元布尔函数 B_α^n(不必用 n 时, 简写为 B_α) 称作由 α实现 的布尔函数. 这是通过下式来定义的:

$$B_\alpha^n(X_1,\cdots,X_n)=\alpha \text{ 的真值, 当 } \alpha \text{ 中的 } \mathbf{A}_1,\cdots,\mathbf{A}_n \text{ 取值为 } X_1,\cdots,X_n \text{ 时.}$$

换句话说, $B_\alpha^n(X_1,\cdots,X_n)=\bar{v}(\alpha)$, 其中 v 是 $\{\mathbf{A}_1,\cdots,\mathbf{A}_n\}$ 的一个使得 $v(\mathbf{A}_i)=X_i$ 的真值指派. 这样, 当 α 取定时, B_α^n 就可以看作是 v 的函数, 取值为 $\bar{v}(\alpha)$.

例如, 前面列出的布尔函数都可以用这种方式得到:

$$
\begin{aligned}
I_i^n&=B_{\mathbf{A}_i}^n,\\
N&=B_{\neg\mathbf{A}_1}^1,\\
K&=B_{\mathbf{A}_1\wedge\mathbf{A}_2}^2,\\
A&=B_{\mathbf{A}_1\vee\mathbf{A}_2}^2,\\
C&=B_{\mathbf{A}_1\to\mathbf{A}_2}^2,\\
E&=B_{\mathbf{A}_1\leftrightarrow\mathbf{A}_2}^2.
\end{aligned}
$$

对这些函数进行组合可以得到其他的函数. 例如, 46

$$B_{\neg\mathbf{A}_1\vee\neg\mathbf{A}_2}^2(X_1,X_2)=A(N(I_1^2(X_1,X_2)),N(I_2^2(X_1,X_2))).$$

(可以把等式的右边与 $\neg\mathbf{A}_1\vee\neg\mathbf{A}_2$ 的波兰记法进行比较.) 稍后, 我们讨论每个布尔函数是不是都可以这样得到.

正如下面的定理所述, 如果将注意力从合式公式转移到它们实现的布尔函数上, 我们能够有效地确定重言等价的合式公式. 现在给集合 $\{F,T\}$ 排一个顺序: $F<T$. (如果 $F=0$, $T=1$, 则这是一个很自然的顺序.)

定理 15A　设 α 和 β 是合式公式，它们的命题符号在 $\mathbf{A}_1, \cdots, \mathbf{A}_n$ 中. 那么

(a) $\alpha \models \beta$ 当且仅当对所有的 $\overrightarrow{X} \in \{F,T\}^n, B_\alpha(\overrightarrow{X}) \leqslant B_\beta(\overrightarrow{X})$.

(b) $\alpha \models\!\!\dashv \beta$ 当且仅当 $B_\alpha = B_\beta$.

(c) $\models \alpha$ 当且仅当 B_α 是具有真值 T 的常函数.

证明　(a) $\alpha \models \beta$ 当且仅当对于所有 2^n 个对 $A_1, \cdots, A_n$ 的真值指派中的任意一个 v，只要 $\bar{v}(\alpha) = T$，就有 $\bar{v}(\beta) = T$. (即使 α 和 β 中的命题符号没有包含所有的 $A_1, \cdots, A_n$，这也是正确的；见 1.2 节中的习题 6.) 这样，就有

$$\begin{aligned} \alpha \models \beta \quad & \text{iff} \quad \text{对所有的 } 2^n \text{ 个指派 } v,\ \bar{v}(\alpha) = T \ \Rightarrow\ \bar{v}(\beta) = T\,; \\ & \text{iff} \quad \text{对所有的 } 2^n \text{ 个 } n\text{- 元组 } \overrightarrow{X},\ B_\alpha^n(\overrightarrow{X}) = T \ \Rightarrow\ B_\beta^n(\overrightarrow{X}) = T; \\ & \text{iff} \quad \text{对所有的 } 2^n \text{ 个 } n\text{- 元组 } \overrightarrow{X},\ B_\alpha^n(\overrightarrow{X}) \leqslant B_\beta^n(\overrightarrow{X}). \end{aligned}$$

其中，$F < T$. ■

除了要确定重言等价的合式公式外，我们还可以脱离形式语言，自由地考虑任意的布尔函数是不是由合式公式实现的. 当然，这种自由只是表面上的：

定理 15B　设 G 是一个 n 元布尔函数，$n \geqslant 1$. 可以找到一个合式公式 α 使得 $G = B_\alpha^n$，即 α 实现函数 G.

引入布尔函数的根本目的是为了把这个定理公式化，该定理是由 Emil Post 在 1921 年提出的.

证明　情形 I：G 是具有真值 F 的常值函数. 只要取 $\alpha = \mathbf{A}_1 \wedge \neg\, \mathbf{A}_1$.

情形 II：否则就有 k 个点，G 在这些点上具有真值 $T, k > 0$，即

$$\begin{aligned} \overrightarrow{X}_1 &= \langle X_{11}, X_{12}, \cdots, X_{1n} \rangle, \\ \overrightarrow{X}_2 &= \langle X_{21}, X_{22}, \cdots, X_{2n} \rangle, \\ &\cdots \\ \overrightarrow{X}_k &= \langle X_{k1}, X_{k2}, \cdots, X_{kn} \rangle, \end{aligned}$$

47　令

$$\beta_{ij} = \begin{cases} \mathbf{A}_j & \text{iff} \quad X_{ij} = T, \\ (\neg\, \mathbf{A}_j) & \text{iff} \quad X_{ij} = F, \end{cases}$$

$$\gamma_i = \beta_{i1} \wedge \cdots \wedge \beta_{in},$$

$$\alpha = \gamma_1 \vee \gamma_2 \vee \cdots \vee \gamma_k.$$

我们要证明 $G = B_\alpha^n$.

这里来看一个具体的例子，这对我们的证明是有帮助的. 设 G 是如下的三元布尔函数：

$$G(F,F,F) = F,$$

$$G(F,F,T) = T,$$

$$G(F,T,F) = T,$$

$$G(F,T,T)=F,$$
$$G(T,F,F)=T,$$
$$G(T,F,T)=F,$$
$$G(T,T,F)=F,$$
$$G(T,T,T)=T.$$

使 G 取真值 T 的三元组有 4 个：

$$\begin{array}{ll} FFT & \neg\, \mathbf{A}_1 \wedge \neg\, \mathbf{A}_2 \wedge \mathbf{A}_3, \\ FTF & \neg\, \mathbf{A}_1 \wedge \mathbf{A}_2 \wedge \neg\, \mathbf{A}_3, \\ TFF & \mathbf{A}_1 \wedge \neg\, \mathbf{A}_2 \wedge \neg\, \mathbf{A}_3, \\ TTT & \mathbf{A}_1 \wedge \mathbf{A}_2 \wedge \mathbf{A}_3. \end{array}$$

每个三元组的右边是相应的合取式 γ_i，那么 α 就可以写成如下的式子：

$$(\neg\, \mathbf{A}_1 \wedge \neg\, \mathbf{A}_2 \wedge \mathbf{A}_3) \vee (\neg\, \mathbf{A}_1 \wedge \mathbf{A}_2 \wedge \neg\, \mathbf{A}_3) \vee$$
$$(\mathbf{A}_1 \wedge \neg\, \mathbf{A}_2 \wedge \neg\, \mathbf{A}_3) \vee (\mathbf{A}_1 \wedge \mathbf{A}_2 \wedge \mathbf{A}_3).$$

注意：α 显式地列出 G 取真值 T 的所有三元组.

返回到定理的证明，注意到对于 $1 \leqslant i \leqslant k$，有 $B_\alpha^n(\overrightarrow{X}_i) = T$. (由于对应于 $\overrightarrow{X}_i$ 的真值指派满足 γ_i，因此也满足 α.) 另一方面，只有一个 $\mathbf{A}_1, \cdots, \mathbf{A}_n$ 的真值指派能够满足 γ_i，因而只有 k 个这样的真值指派满足 α. 故对 $2^n - k$ 个其他的 n 元组有 $B_\alpha^n(\overrightarrow{Y}) = F$. 这样在所有情况下，都有 $B_\alpha^n(\overrightarrow{Y}) = G(\overrightarrow{Y})$. ■

由此定理可知，每个布尔函数都是可以实现的. 当然，实现函数 G 的 α 不是唯一的，任意重言等价的合式公式都能实现同一个函数. 有时，要尽可能选择最短的 α. (在前面的例子中，合式公式

$$\mathbf{A}_1 \leftrightarrow \mathbf{A}_2 \leftrightarrow \mathbf{A}_3$$

也能够实现 G.)

48

上面定理的一个推论是我们已经有了足够多的命题联结符号 (实际上，已经过多了). 假定我们通过添加一些新的命题联结词 (如本节开始的多数决定联结词) 扩展形式语言，在这种扩展后的语言中任何一个合式公式 φ 实现一个布尔函数 B_φ^n. 根据上述定理，在原来的语言中有一个合式公式 α 使得 $B_\varphi^n = B_\alpha^n$. 因此由定理 15A 知，φ 与 α 重言等价.

事实上，定理的证明说明 α 具有一种特殊的形式. 首先，在 α 中只出现了 3 个联结符号：$\wedge$，$\vee$，$\neg$. 同时，α 的这种形式称作 *析取范式* (缩写为 DNF)，也就是说，α 是一个析取式：

$$\alpha = \gamma_1 \vee \cdots \vee \gamma_k,$$

其中每个 γ_i 是合取式：

$$\gamma_i = \beta_{i1} \wedge \cdots \wedge \beta_{in_i}$$

并且每个 β_{ij} 都是命题符号或者命题符号的否定.

(合式公式的析取范式的优点在于其显式地给出了满足公式的真值指派.)这样就有以下结论:

推论 15C 对于任意 (可满足) 合式公式 φ, 可以找到一个与其重言等价的析取范式 α.

由于每个函数 $G:\{F,T\}^n \to \{F,T\}, n \geqslant 1$, 都可以由一个只使用 $\{\wedge,\vee,\neg\}$ 中的联结词的合式公式实现, 我们称 $\{\wedge,\vee,\neg\}$ 是完备的. (事实上, 完备性是与这些联结词符号对应的布尔函数 K, A 和 N 的一种性质. 不过, 上述说法是方便的.) 一旦有了完备的联结词的集合, 任何合式公式都重言等价于一个合式公式, 其联结词都出自这个集合中. (对于给定的合式公式 φ, 可以通过使用这些联结词来找到 α, 实现 B_φ, 这样 $\alpha \models\models \varphi$). $\{\wedge,\vee,\neg\}$ 的完备性可以进一步推广为如下定理:

定理 15D $\{\neg,\wedge\}$ 和 $\{\neg,\vee\}$ 都是完备的.

证明 我们要证明所有的布尔函数 G 可以由仅使用 $\{\wedge,\neg\}$ 中联结词的合式公式实现. 首先从可以实现 G 的合式公式 α 开始, 这个合式公式只使用 $\{\wedge,\vee,\neg\}$ 中的联结词. 我们只需要找到一个与 α 重言等价并且只使用了 $\{\wedge,\neg\}$ 中的联结词的合式公式 α' 就可以了. 对于这一点, 使用德摩根律可得:

$$\beta \vee \gamma \models\models \neg(\neg\beta \wedge \neg\gamma).$$

49 反复使用该运算律, 就可以将 α 中的 $\vee$ 完全消去.

(规范地, 通过对 α 使用归纳法可以证明, 存在一个重言等价的 α', α' 仅使用了联结词 $\wedge$, $\neg$. 归纳步骤中有两种情形:

$\neg$ 的情形: 如果 α 是 $\neg\beta$, 则令 $\alpha' = (\neg\beta')$.

$\vee$ 的情形: 如果 α 是 $(\beta\vee\gamma)$, 则令 $\alpha' = \neg(\neg\beta' \wedge \neg\gamma')$, 由于 β' 与 γ' 分别与 β, γ 重言等价, 因此

$$\begin{aligned} \alpha' &= \neg(\neg\beta' \wedge \neg\gamma') \\ &\models\models \neg(\neg\beta \wedge \neg\gamma) \\ &\models\models \beta\vee\gamma \\ &= \alpha. \end{aligned}$$

在以后关于联结词完备的证明中, 这个归纳将被略去. 比如, 我们将只给出用 $\neg$ 和 $\wedge$ 来表示 $\vee$ 的方法. ■

证明一个联结词集合的不完备性往往比证明其完备性更困难. 基本的方法是首先 (通常使用归纳) 证明对于仅使用所给集合中的联结词的任意合式公式 α, 函数 $B_\alpha^n d$ 都具有某种特性, 然后证明存在一个布尔函数不具有这种特性.

例 $\{\wedge,\to\}$ 是不完备的.

证明 基本思想是, 如果所有的命题符号的赋值都是 T, 那么使用这些联结词的公式的值就是 T. 特别地, 不存在与 $\neg\mathbf{A}$ 重言等价的合式公式.

详细地说, 通过归纳法可以证明对于仅使用这些联结词的任意合式公式 α, 如果 α 中只用到命题符号 $\mathbf{A}$, 我们就有 $\mathbf{A} \models \alpha$. (从函数的角度来看, 即 $B_\alpha^1(T) = T$.) 类似地可以证明 $\{\wedge,\vee,\neg,\leftrightarrow\}$ 也是不完备的. ■

对每个 n, 存在 2^{2^n} 个 n 元布尔函数. 因而, 如果我们把一个联结词和它的布尔函数看成是一样的 (如前所述的 $\wedge$ 及函数 K), 那么就有 2^{2^n} 个 n 元联结词. 这里我们列出 $n \leqslant 2$ 的所有情况.

1.5.1 0 元联结词

有两个 0 元布尔函数: F 和 T. 相应的联结符号是 $\perp$ 和 $\top$. n 元联结词符号将 n 个合式公式连接成一个新的合式公式. 当 $n = 0$ 时, $\perp$ 本身是一个合式公式, 由于对每个 v, 有 $\bar{v}(\perp) = F$, 所以它不同于任意的命题符号. 即 $\perp$ 是一个总是赋值为 F 的逻辑符号. 类似地, $\top$ 也是一个合式公式, 对每个 v, 有 $\bar{v}(\top) = T$. 于是, $\mathbf{A} \to \perp$ 就是一个合式公式, 重言等价于 $\neg\mathbf{A}$, 这可以从一个两行的真值表得出.

50

1.5.2 一元联结词

有 4 个一元联结词, 但是只有一个有意义, 即否定联结词. 其余的 3 个一元布尔函数是恒等函数和两个常值函数.

1.5.3 二元联结词

有 16 个二元联结词, 但是只有表 1-6 中列出的 10 个是真正 "二元" 的.

表 1-6

符号	等价式	说明
	$\top$	二元常数, 本质上是 0 元的
	$\perp$	二元常数, 本质上是 0 元的
	$\mathbf{A}$	投影, 本质上是 1 元的
	$\mathbf{B}$	投影, 本质上是 1 元的
	$\neg\mathbf{A}$	否定, 本质上是 1 元的
	$\neg\mathbf{B}$	否定, 本质上是 1 元的
$\wedge$	$\mathbf{A} \wedge \mathbf{B}$	与; 如果 $F = 0, T = 1$, 这给出的是域 $\{0, 1\}$ 上的乘法运算
$\vee$	$\mathbf{A} \vee \mathbf{B}$	或
$\to$	$\mathbf{A} \to \mathbf{B}$	蕴涵
$\leftrightarrow$	$\mathbf{A} \leftrightarrow \mathbf{B}$	等价
$\leftarrow$	$\mathbf{A} \leftarrow \mathbf{B}$	逆蕴涵
$+$	$(\mathbf{A} \vee \mathbf{B}) \wedge \neg(\mathbf{A} \wedge \mathbf{B})$	排他或, "$\mathbf{A}$ 或 $\mathbf{B}$ 且不能同时成立"; 如果 $F = 0, T = 1$, 这给出的是域 $\{0, 1\}$ 上的模 2 加法运算
$\downarrow$	$\neg(\mathbf{A} \vee \mathbf{B})$	或非, "非 $\mathbf{A}$ 且非 $\mathbf{B}$"
$\mid$	$\neg(\mathbf{A} \wedge \mathbf{B})$	与非, "$\mathbf{A}$ 与 $\mathbf{B}$ 不能同时成立"; 称为 Sheffer 竖
$<$	$(\neg\mathbf{A}) \wedge \mathbf{B}$	通常的排序, 其中 $F < T$
$>$	$\mathbf{A} \wedge (\neg\mathbf{B})$	通常的排序, 其中 $F < T$

1.5.4 三元联结词

有 256 个三元联结词, 但从本质上看, 2 个是 0 元的, $6(=2\cdot\binom{3}{1})$ 个是一元的, $30(=10\cdot\binom{3}{2})$ 个是二元的. 因此只剩下 218 个是事实上的三元联结词. 前面出现的多数决定联结词 # 就是一个. 类似地, 还有少数决定联结词. 习题 7 给出 $+^3$, 即三元模 2 加法运算. $+^3\alpha\beta\gamma$ 取值

为 T 当且仅当 α, β, γ 中有奇数个的赋值为 T. 这个公式等价于 $\alpha+\beta+\gamma$ 和 $\alpha \leftrightarrow \beta \leftrightarrow \gamma$.
51 在习题 8 中给出了另外一个三元联结词.

例 $\{|\}$ 和 $\{\downarrow\}$ 是完备的.

证明 对于 $|$,

$$\neg\alpha \models\dashv \alpha|\alpha$$
$$\alpha\vee\beta \models\dashv (\neg\alpha)|(\neg\beta).$$

由于 $\{\neg,\vee\}$ 是完备的, 并且 $\neg,\vee$ 可以用 $|$ 表示, 故 $\{|\}$ 是完备的. ■

例 $\{\neg,\rightarrow\}$ 是完备的. 事实上, 在 10 个真正的二元联结词中有 8 个能够和 $\neg$ 构成完备集合. 其余的两个是 $+$ 和 $\leftrightarrow$, 见习题 5.

例 $\{\perp,\rightarrow\}$ 是完备的. 事实上, 使用这两个联结词甚至可以实现 0 元布尔函数, 它们可以说是超完备的.

习题

1. 设 G 是三元布尔函数, 定义如下:

$$G(F,F,F)=F, \quad G(T,F,F)=T,$$
$$G(F,F,T)=T, \quad G(T,F,T)=F,$$
$$G(F,T,F)=T, \quad G(T,T,F)=F,$$
$$G(F,T,T)=F, \quad G(T,T,T)=F.$$

   (a) 给出一个仅使用 $\vee$, $\wedge$ 和 $\neg$ 的能够实现 G 的合式公式.
   (b) 给出一个合式公式, 其联结词最多出现 5 个.
2. 证明 $|$ 和 $\downarrow$ 是仅有的两个自我完备的二元联结词.
3. 证明 $\{\neg,\#\}$ 是不完备的.
4. 设 M 是三元少数决定联结词, ($\bar{v}(\mathrm{M}\alpha\beta\gamma)$ 总是与 $\bar{v}(\alpha)$, $\bar{v}(\beta)$ 和 $\bar{v}(\gamma)$ 中的多数取值不同.) 证明
   (a) $\{\mathrm{M},\perp\}$ 是完备的.
   (b) $\{\mathrm{M}\}$ 是不完备的.
5. 证明 $\{\top,\perp,\neg,\leftrightarrow,+\}$ 是不完备的. 提示: 证明任意使用这些联结词和命题符号 $\mathbf{A}$, $\mathbf{B}$ 的合式公式在 $\bar{v}(\alpha)$ 的 4 种可能的取值下有偶数个取值为 T.
   说明: 另一种方法是使用域 $\{0,1\}$ 上的代数, 任意使用这些联结词的可实现的布尔函数具有线性的
52 特性.
6. 证明 $\{\wedge,\leftrightarrow,+\}$ 是完备的, 且没有完备的真子集.
7. 设 $+^3$ 是三元联结词, 满足 $+^3\alpha\beta\gamma$ 等价于 $\alpha+\beta+\delta$.
   (a) 试证明 $\{\top,\perp,\wedge,+^3\}$ 是完备的.
   (b) 证明这个集合没有完备的真子集.
   说明: $+^3$ 是三元 奇偶 联结词, $\bar{v}(+\alpha_1\alpha_2\alpha_3)=T$ 的条件是有 奇数 个 α_i 使得 $\bar{v}(\alpha_i)=T$. $+$ 是二元奇偶联结词. 在定理 15B 的证明中出现的函数 G 是三元奇偶函数.
8. 设 $\mathbb{I}$ 是一个三元联结词, 满足条件: $\mathbb{I}\alpha\beta\gamma$ 取真值当且仅当 α, β, γ 中恰有一个取真值. 证明不存在二元联结词 $\circ$ 和 $\triangle$, 使得 $\mathbb{I}\alpha\beta\gamma$ 等价于 $(\alpha\circ\beta)\triangle\gamma$.

9. 称 α 为合取范式 (缩写为 CNF) 当且仅当它是一个合取式：

$$\alpha = \gamma_1 \wedge \cdots \wedge \gamma_k$$

其中每一个 γ_i 都是析取式：

$$\gamma_i = \beta_{i1} \vee \cdots \vee \beta_{in}$$

且每个 β_{ij} 都是命题符号或者命题符号的否定.

(a) 试给出与 $\mathbf{A} \leftrightarrow \mathbf{B} \leftrightarrow \mathbf{C}$ 重言等价的一个合取范式.

(b) 证明对任意 (不恒真的) 公式都存在一个重言等价的合取范式.

10. 将 0 元联结词 $\perp, \top$ 加到形式语言中, 对每个合式公式 φ 和命题符号 A, 令 $\varphi_{\top}^{A}$ 是一个将 φ 中的 A 置换为 $\top$ 后得到的合式公式. 类似地, 可以定义 $\varphi_{\perp}^{A}$. 那么令 $\varphi_{*}^{A} = (\varphi_{\top}^{A} \vee \varphi_{\perp}^{A})$. 证明

(a) $\varphi \models \varphi_{*}^{A}$.

(b) 如果 $\varphi \models \psi$ 且 ψ 中没有出现 A, 则 $\varphi_{*}^{A} \models \psi$.

(c) 公式 φ 是可满足的当且仅当 φ_{*}^{A} 是可满足的.

说明：可以将 φ_{*}^{A} 看作与 φ 的含义相同, 但没有使用符号 A. (a)(b) 两条说明 φ_{*}^{A} 是 φ 不使用 A 的最强结果. 一般地, φ_{*}^{A} 与 φ 不是重言等价的, 但是由 (c) 可知二者是“等价可满足的”. 由 φ 得到 φ_{*}^{A} 的运算称为 *消去A*.

11. (插值定理) 如果 $\alpha \models \beta$, 那么存在某个 γ, 它所含的命题符号既出现在 α 中也出现在 β 中, 使得 $\alpha \models \gamma \models \beta$. *提示*：使用前面习题的结论.

说明：习题 11 和一阶逻辑中的结论很类似, 但是其证明却大不相同, 因为一阶逻辑中没有与习题 10 相似的结论.

53

12. $\{\perp, \top, \wedge\}$ 是完备的吗？给出理由.

1.6 交换电路[1]

考虑具有 n 个输入和 1 个输出的电子设备 (传统的黑盒子), 如图 1-1 所示. 假定每个输入和输出信号都是两个值中的一个, 这两个可能的取值称之为 F 和 T. (值 F 也被定义为 0 电位, 而在选择适当的电压单位后, 值 T 的值也可以看作 1 电位.) 还假设设备没有记忆功能, 即当前的输出电位仅依赖于当前的输入 (与先前的输入无关). 这样, 电子设备的功能就可以使用布尔函数来描述：

$$G(X_1, \cdots, X_n) = \text{在给定输入信号}X_1, \cdots, X_n\text{时的输出电位}$$

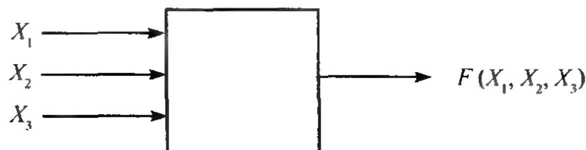


图 1-1 具有 3 个输入的电子设备

满足所有这些假设的电子设备组成了电子计算机电路的核心. 例如, 有两个输入的与门, 其输出是两个输入的最小值 (其中 $F < T$), 该设备实现了上节中的布尔函数 K. 为方便起见, 我们把两个输入分别记为 $\mathbf{A}_1$, $\mathbf{A}_2$, 输出记作 $\mathbf{A}_1 \wedge \mathbf{A}_2$.

1. 本节内容是关于前面几节内容的应用, 可以跳过本节继续学习后面的内容.

类似地，我们可以为其他的命题联结词设计电子设备．对两个输入的或门，其输出为输入电压的最大值 (如图 1-2 所示)．对应于否定联结词的是一个 NOT 设备 (非门变极器)，其输出为输入的相反电压．

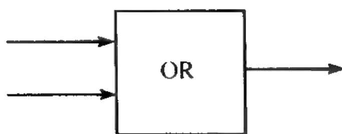


图 1-2　或门

54　电路可以由许多这种类型的设备构成．自然地，我们可以使用形式语言中的合式公式来标记不同点的电压 (如图 1-3 所示)．相反地，给定了与输出对应的合式公式，我们就可以精确地重建电路，这有点类似于合式公式的生成树．

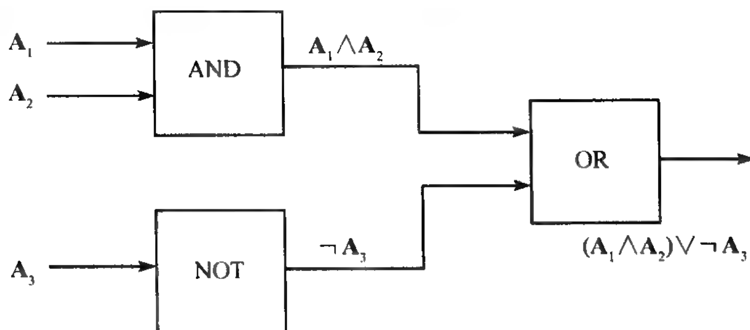


图 1-3　带合式公式标记的电路

例如，$((\mathbf{A} \wedge \mathbf{B}) \wedge \mathbf{D}) \vee ((\mathbf{A} \wedge \mathbf{B}) \wedge \neg \mathbf{C})$ 对应的电路如图 1-4 所示．通常，我们不希望 $\mathbf{A} \wedge \mathbf{B}$ 代表的电路重复使用．

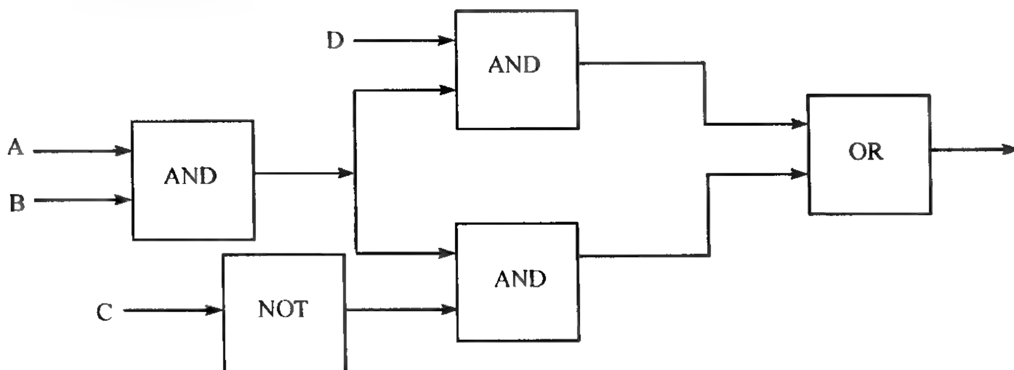


图 1-4　$((\mathbf{A} \wedge \mathbf{B}) \wedge \mathbf{D}) \vee ((\mathbf{A} \wedge \mathbf{B}) \wedge \neg \mathbf{C})$ 对应的电路

55　重言等价的合式公式产生的电路最终具有相同的功能，但是其代价和速度 (若运算中设备并非总是处于待用状态) 可能会大不相同．电路的延迟 (也称为电路的深度) 定义为信号从一个输入端到一个输出端所经过的最多的盒子数．与公式相对应的延迟概念可以定义为

(1) 命题符号的延迟为 0.

(2) $\neg \alpha$ 的延迟比 α 的延迟大 1.

(3) $\alpha \wedge \beta$ 的延迟比 α 与 β 的延迟中最大的一个大 1.

类似地，可以规定其他联结词的情况.

例如，$(\mathbf{A}_1 \wedge \mathbf{A}_2) \vee \neg\, \mathbf{A}_3$ 的电路使用 3 个设备，其延迟为 2；其重言等价的公式 $\neg(\mathbf{A}_3 \wedge (\neg\mathbf{A}_1 \vee \neg\mathbf{A}_2))$ 对应的电路需要 5 个设备，延迟为 4. 计算机工程师所面临的问题就是：对于给定的电路 (或者它的合式公式)，要找到一个能满足条件的代价最小的等价电路 (或者重言等价的合式公式)，比如满足最大允许的延迟. 对于这个问题，有一组可使用的设备；例如，可以使用

NOT, 2 输入与门, 3 输入或门

(我们当然希望可用的设备与一个完备的联结词集合相对应.) 这组设备决定了一种形式语言，其中每个设备对应一个联结符号.

例 1 输入：**A**, **B**, **C**. 输出：**A**, **B**, **C** 中取值最多的值. 可用的设备有：2 输入或门，2 输入与门. 一个解决方法是：

$$((\mathbf{A} \wedge \mathbf{B}) \vee (\mathbf{A} \wedge \mathbf{C}) \vee (\mathbf{B} \wedge \mathbf{C}),$$

它用到了 5 个设备，延迟为 3. 但是一个更好的方案是

$$(\mathbf{A} \wedge (\mathbf{B} \vee \mathbf{C})) \vee (\mathbf{B} \wedge \mathbf{C}),$$

这个分方案用到了 4 个设备，延迟为 3. 没有只使用 3 个设备就能解决这个问题的方法. 见习题 1.

例 2 输入：**A**, **B**. 输出：如果两个输入相同，输出 T；否则，输出 F. 也就是说，这个电路用于鉴别输入是否相同. 可用的设备是：2 输入 NOR. 一个方法是

$$((\mathbf{A} \downarrow \mathbf{A}) \downarrow \mathbf{B}) \downarrow ((\mathbf{B} \downarrow \mathbf{B}) \downarrow \mathbf{A}).$$

其用了 5 个设备，有没有更好的方法呢？更深入的问题是：有没有有效的寻找最小方案的方法？在此，我们仅仅是提出这些问题而已. 近几年来已经有大量的工作来深入研究这类问题.

56

例 3 (继电器开关) 输入：**A**, ¬**A**, **B**, ¬**B**, ⋯. 设备：或门 (任意个输入)，与门 (任意个输入). 代价：设备空闲，每次使用一个输入需要一个单位. 要验证 **A** 与 **B** 是否相等，我们需要使用

$$(\mathbf{A} \wedge \mathbf{B}) \vee (\neg\, \mathbf{A} \wedge \neg\, \mathbf{B}).$$

电路的接线图如图 1-5 所示. 电路中有电流通过当且仅当 **A** 与 **B** 被赋以相同的值. (这个公式等价于 $\mathbf{A} \leftrightarrow \mathbf{B}$，只要一个变量的值改变，其真值就会改变. 因此，该电路用于走廊灯的双控开关.)

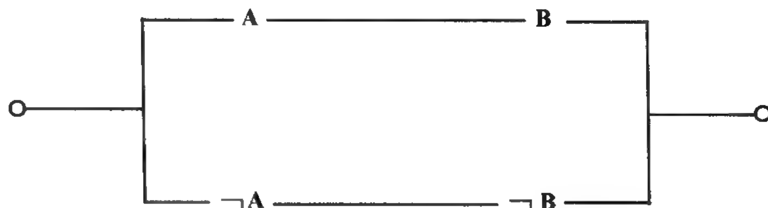
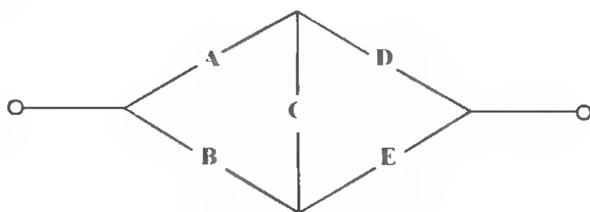


图 1-5 $(\mathbf{A} \wedge \mathbf{B}) \vee (\neg\, \mathbf{A} \wedge \neg\, \mathbf{B})$ 的接线图

然而，需要注意的是，继电器开关不适合于本节开始所述的情形. 继电器是一个双边设备，电流可以从任何一个方向通过. 这个特点可用于“桥”电路设计 (如图 1-6 所示). 此处所述的方法不用于此类电路.



57

图 1-6　桥电路

例 4　有 4 个输入，电路用于实现布尔函数 G, G 在下述情况下取真值 T: $\langle F,F,F,T\rangle$, $\langle F,F,T,F\rangle$, $\langle F,F,T,T\rangle$, $\langle F,T,F,F\rangle$, $\langle F,T,F,T\rangle$, $\langle F,T,T,F\rangle$, $\langle F,T,T,T\rangle$, $\langle T,F,F,T\rangle$. 在以下情况下取真值 F: $\langle T,F,F,F\rangle$, $\langle T,F,T,F\rangle$, $\langle T,T,F,F\rangle$, $\langle T,T,T,F\rangle$, $\langle T,T,T,T\rangle$. 在其余的情况下 ($\langle F,F,F,F\rangle$, $\langle T,F,T,T\rangle$, $\langle T,T,F,T\rangle$), 我们不关心 G 的取值. (电路的应用使得这 3 种组合是不可能发生的.)

我们知道 G 是利用 $\{\wedge,\vee,\neg\}$ 实现的，但是我们希望以一种有效的方式实现. 第一步是以一种更易于理解的方式表示数据，如图 1-7 的表示. 由于 $G(F,F,F,T)=T$, 我们在坐标为 ($\neg$ **A**, $\neg$ **B**, $\neg$**C**, **D**) 的方格内填入 T, 类似地，由于 $G(T,T,F,F)=F$, 我们在坐标为 (**A**, **B**, $\neg$**C**, $\neg$ **D**) 的方格内填入 F. 我们不关心的 3 个方格，使其保持空白.

	¬A, ¬B	¬A, B	A, B	A, ¬B	
¬C		T	F	F	¬D
¬C	T	T			D
C	T	T	F		D
C	T	T	F	F	¬D

图 1-7　例 4 图表

现在我们来寻找一种简单的几何模式，阴影区域包括了所有的 T 和非 F 的方格. 对应的公式为

$$(\neg\ \mathbf{A})\vee(\neg\ \mathbf{C}\wedge\mathbf{D}),$$

58

这是符合我们的所有要求的相对简单的公式，注意 **B** 的输入根本就是可有可无的.

习题

1. 验证在本节例 1 中不存在仅使用 3 个设备的方案.
2. 一个文字是指一个合式公式，该合式公式是一个命题符号或者命题符号的否定. φ 的一个*蕴涵元*是指文字 (使用不同的命题符号) 的合取式 α, 它满足 $\alpha\models\varphi$. 我们已经在 1.5 节中证明了 (参看推论

15C) 任何可满足的合式公式φ都重言等价于析取式 $\alpha_1 \vee \cdots \vee \alpha_n$，其中每个 α_i 是 φ 的蕴涵元. 如果去掉 φ 的蕴涵元中的任何一个文字，它就不再是 φ 的蕴涵元了，那么我们称这样的蕴涵元是基本的. 任何蕴涵元的析取如果具有最短的长度，并且等价于 φ，那么它只包含基本的蕴涵元.

(a) 找出下式所有的基本蕴涵元：

$$(\mathbf{A} \to \mathbf{B}) \wedge (\neg\, \mathbf{A} \to \mathbf{C}).$$

(b) 重言等价于上述公式的基本蕴涵元的析取式有哪些？

3. 对下列公式重复习题 2 的 (a) 与 (b)：

$$(\mathbf{A} \vee \neg\, \mathbf{B}) \wedge (\neg\, \mathbf{C} \vee \mathbf{D}) \to \mathbf{B} \wedge ((\mathbf{A} \wedge \mathbf{C}) \vee (\neg\, \mathbf{C} \wedge \mathbf{D})).$$

1.7 紧致性和能行性

1.7.1 紧致性

现在我们给出 1.2 节中提到的紧致性定理的证明. 合式公式的集合 Σ 称为是*可满足的*当且仅当有一个真值指派满足其中的每个合式公式.

紧致性定理 合式公式的集合是可满足的当且仅当它的每个有限子集是可满足的.

我们暂且把 Σ 的每个有限子集是可满足的，简称为 Σ 是**有限可满足的**. 紧致性定理说的就是这个概念与可满足性是一致的. 请注意：如果 Σ 是可满足的，则它的每个有限子集自然也是可满足的. 同时，如果 Σ 是有限的，逆命题显然也成立 (因为每个集合都是自身的子集). 我们要证明的就是当一个无限子集是有限可满足时，它也是可满足的.

紧致性定理的证明 证明包括两部分. 在第一部分中，我们考虑有限可满足的集合 Σ，将其扩展到最大可满足的集合 Δ；在第二部分中，我们利用 Δ 给出一个能够满足 Σ 的真值指派.

第一部分：设 $\alpha_1, \alpha_2, \cdots$ 是合式公式的一个枚举. (这是可以做到的，因为命题集合是可数的，表达式集合也是可数的；见定理 0B.) 我们做如下的递归定义：

$$\Delta_0 = \Sigma,$$

$$\Delta_{n+1} = \begin{cases} \Delta_n; \alpha_{n+1} & \text{如果是有限可满足的} \\ \Delta_n; \neg\, \alpha_{n+1} & \text{其他情况} \end{cases}$$

(*提示*：$\Delta_n; \alpha_{n+1} = \Delta_n \cup \{\alpha_{n+1}\}$) 那么每个 Δ_n 是有限可满足的，见习题 1. 记 Δ_n 的极限 $\Delta = \bigcup_n \Delta_n$.

显然，(1) $\Sigma \subseteq \Delta$. (2) 对任意的合式公式 α，要么 $\alpha \in \Delta$ 要么 $(\neg\, \alpha) \in \Delta$. (3) Δ 是有限可满足的. 因为任意有限子集肯定是某个 Δ_n 的子集，因此也是可满足的.

以上是第一部分的证明. 现在我们有了一个集合 Δ 具有上述性质 (1)~(3). 通常，这样的集合不是唯一的，但至少有一个存在. (我们可以使用另外一种方法来证明这样的集合 Δ 是存在的，这种方法使用了佐恩引理. 即使在命题符号是不可数的情况下，这种方法仍然可行. 熟悉佐恩引理的读者可以感受一下这种可行性.)

第二部分：我们定义一个由所有命题符号组成的集合上的真值指派 v：对任意的命题符号 A, $v(A) = T$ iff $A \in \Delta$．那么对于任意的合式公式，可以证明 v 满足 φ iff $\varphi \in \Delta$. 这一点可以对 φ 使用归纳法进行证明，见习题 2．由于 $\Sigma \subseteq \Delta$, v 一定能够满足 Σ 的每个合式公式. ■

推论 17A 如果 $\Sigma \models \tau$, 那么存在有限集合 $\Sigma_0 \subseteq \Sigma$ 使得 $\Sigma_0 \models \tau$.

证明 很明显，$\Sigma \models \tau$ 当且仅当 $\Sigma; \neg \tau$ 是不可满足的.

对每个有限的 $\Sigma_0 \subseteq \Sigma$, $\Sigma_0 \not\models \tau$

$\Rightarrow$ 对每个有限的 $\Sigma_0 \subseteq \Sigma$, $\Sigma_0; \neg \tau$ 是可满足的

$\Rightarrow$ $\Sigma; \neg \tau$ 是有限可满足的

$\Rightarrow$ $\Sigma; \neg \tau$ 是可满足的

$\Rightarrow$ $\Sigma \not\models \tau$. ■

60

事实上，上述的推论等价于紧致性定理，见习题 3.

1.7.2 能行性及可计算性

尽管真值表的方法使用起来非常麻烦，但是这种方法还是产生了一些有趣的理论结果．假定有一个合式公式的集合 Σ 和一个合式公式 τ，那么有没有能行的过程来确定 $\Sigma \models \tau$ 是否成立．所谓能行的过程必须满足下面的条件：

(1) 必须有确切的指令用以解释如何执行这个过程，指令长度必须是有限的．指令应该能够以某种可能的方式传达给执行计算任务的人或者机器，我们无法给予执行任务者一个无限的对象，但我们也不能给指令强加一个准确的长度上限．如果指令比宇宙中的电子还多，那么我们也只能耸耸肩膀说：“这个程序太长了．”

(2) 从执行指令者的角度出发，这些指令无需精彩或者有创意．勤奋的办事员 (不懂数学，但是能够很好地执行命令) 或计算机 (根本不会思考) 都能够执行这些指令．也就是说，这些指令必须能够被 *机械地执行*．这样的方法必须要避免使用随机设备 (如投掷硬币) 或只能近似计算的设备．

(3) 如上所述，对于给定的合式公式 τ，判定过程必须能够在有限步内给出“是”或“否”的答案．(即判定过程就是一个确定答案的 *算法*．)

另一方面，在答案出现之前，我们无法给出可能需要的时间，也不能事先知道所需要的草稿纸 (或者其他存储介质) 的数量．在不考虑其他情况的前提下，这些要依赖于输入 τ．然而对任意的输入 τ，判定过程能够在有限步内产生结果，因此所需要的草稿纸的数量肯定是有限的．而执行无限长的步骤来产生结果是不可能的．

如果计算机能够在合理的时间范围内完成一个过程，那么使用电子计算机的人会认为这个过程是能行的．当然，问题在于合理与否是随着环境而改变的．随着计算机内存的增加，其速度也越来越快，那么合理的时间范围能行的过程也会随之增加．我们需要一个有效方法的概念，这个概念是在不考虑运行时间和存储空间之类的限制的情况下给出的．

61

当然，上述关于能行的概念是不精确的．事实上，在本书中该词仅以非正规的和直观的方式出现．(在第 3 章，我们会遇到一个准确的概念：递归) 但是，只要我们承认确实存在特定的能行过程，这种非正式的说法也就足够了．我们只要完整地列出该过程，证明它是可行

的，那么人们就会认为这个过程是能行的．(这依赖于这样的 经验 事实：如果数学家认为这个过程是能行的，那么其他人也有这样的结论．) 可是，如果我们需要否定的结果，即不存在这样的能行过程，那么非正式的说法就不足以说明问题了．(在第 3 章中，我们的确需要一些这样的否定结果．) 由于能行的概念是非正式的，因此与之相关的定义、定理将会标之以星号 *．例如：

***定理 17B** 对于给定的表达式 ε，存在一个能行的判定过程用来确定 ε 是否是合式公式．

证明 见 1.3 节中的算法及其脚注． ■

形式语言具有无穷多个命题符号．由这一事实产生了一个技术问题：当谈及“给定”一个表达式 ε 的时候，我们想像可以一个接一个地将 ε 中的符号列出来；可是要书写出无穷多个符号是难以置信的．为了避免这种情况发生，我们采用如下的“输入输出格式”：比如，只用 5 个符号组成的串 $\mathbf{A}'''''$ 替代 $\mathbf{A}_5$．这样一来形式语言的字母表中就只有 9 个符号了：

$$(,), \neg, \wedge, \vee, \rightarrow, \leftrightarrow, \mathbf{A} \text{ 和 } '.$$

(如果给这 9 个符号标以数字 1~9，所得到的表达式非常类似于计算环境中的表达式，我们仍然可以把 0 作为分隔符．)

从如下定义的角度看，定理 17B 说明合式公式的集合是可判定的．

***定义** 表达式集合 Σ 是可判定的当且仅当对于给定的表达式 α，存在能行的过程来判定 α 是否属于 Σ.

例如，任意的有限集合是可判定的．(指令可以完整地列出集合中的所有有限个合式公式，算法就能够针对合式公式的列表进行验证．) 有些无限集合也是可判定的，但是并非都是如此．一方面，存在含无限多个 (准确地说，是 $2^{\aleph_0}$ 个) 表达式的集合；另一方面，仅有可数多个能行的判定过程，这是因为判定过程完全由 (有限条) 指令确定．我们知道仅存在 $\aleph_0$ 个有限的字母序列，而指令就是有限的字母序列． 62

***定理 17C** 对于给定的有限多个合式公式的集合 $\Sigma;\tau$，存在能行的判定过程判定 $\Sigma \models \tau$ 是否成立．

证明 1.2 节中真值表的方法足以满足要求． ■

由于无限对象无法用任何直接有效的方式给出，所以在该定理中，我们规定 $\Sigma;\tau$ 是有限的．

***推论 17D** 对于有限集合 Σ，Σ 的重言推论的集合是可判定的．特别地，重言式的集合是可判定的．

如果 Σ 是无限集合 (即使是可判定的)，它的重言推论集合也可能是不可判定的 (见第 3 章)．但是，我们可以得到一个稍弱的结果，一定意义上的半可判定性．

我们称一个表达式集合是 *能行可枚举的*，当且仅当存在一个能行的过程以某种顺序列举出 A 中的元素．如果 A 是无限的，那么这个枚举的过程是不会结束的．但是对任意指定的 A 中的元素，最终 (即在有限的时间内) 一定会出现在这个列表中．

为了便于理解，我们给出一种等价的表述.

***定理 17E**　表达式集合 A 是能行可枚举的当且仅当对任意给定的表达式 ε，存在一个能行的判定过程，这个过程能够得到“是”的答案当且仅当 $\varepsilon \in A$.

如果 $\varepsilon \notin A$，方法可能会得出“否”的答案，也可能会一直不停地进行下去，而不会得到任何结果，但是不能给出结果“是”. 这样的一个判定过程被称为是*半可判定的*——只是判定过程的一半.

***定义**　表达式集合 A 是半可判定的，当且仅当对于任意给定的表达式 ε，存在能行的判定过程，这个过程能够回答“是”当且仅当 $\varepsilon \in A$.

这样，定理 17E 说明了一个集合是能行可枚举的当且仅当它是半可判定的.

证明　如果 A 是能行可枚举的，对于给定的任意表达式 ε，我们可以检验能行过程给出的 A 的列表，如果 ε 出现在该列表中，那么回答“是”. (这样，如果 $\varepsilon \notin A$，则永远不会给出答案. 正是这一点保证了 A 是可判定的. 当 ε 不在 A 的前 10^{10} 个元素中时，通常就无法知道是不是还有 $\varepsilon \notin A$——这种情况下，通常我们也就放弃继续检查了——也许在列表中下一个就是 ε.)

63

相反地，假设有一个定理中所述的判定过程，我们希望能够给出 A 的一个列表. 其思想是列举出所有表达式，并将给定的判定过程应用到其中每一个上. 但是，我们必须估算所需要的时间. 我们可以很简单地列出所有表达式：

$$\varepsilon_1, \varepsilon_2, \varepsilon_3, \cdots$$

接下来按照如下步骤进行：

(1) 花 1 分钟检查 ε_1 是否属于 A(使用给定的判定过程).

(2) 花 2 分钟检查 ε_1，再花 2 分钟检查 ε_2.

(3) 花 3 分钟检查 ε_1，3 分钟检查 ε_2，3 分钟检查 ε_3.

依此类推，当然只要产生了“是”的答案，就将这个可以接受的表达式放到输出列表中. 这样，A 的任何元素最终都会出现在该列表中. (一个元素也许会出现无限多次，除非修改上述指令，以避免出现重复的元素.)　■

显然，任何可判定的集合都是半可判定的，因而也是能行可枚举的.

***定理 17F**　表达式组成的一个集合是可判定的当且仅当它和它的补集 (关系到所有的表达式的集合) 都是能行可枚举的.

证明　见习题 8. 该定理有时称为“Kleene 定理”.　■

如果集合 A 与 B 是能行可枚举的，那么 $A \cap B$ 和 $A \cup B$ 都是能行可枚举的 (习题 11). 可判定集合类在并与交的运算下也是封闭的，另外，在补运算下也是封闭的.

一个更本质性的结论如下：

***定理 17G**　如果 Σ 是合式公式的可判定集合，那么它的重言推论的集合是能行可枚举的.

证明 事实上，只要证明 Σ 是能行可枚举就已经足够了；考虑一个枚举的序列： 64

$$\sigma_1, \sigma_2, \sigma_3, \cdots$$

给定任意的合式公式 τ, (可以用真值表) 检验下列各式是否成立：

$$\varnothing \models \tau,$$
$$\{\sigma_1\} \models \tau,$$
$$\{\sigma_1, \sigma_2\} \models \tau,$$
$$\{\sigma_1, \sigma_2, \sigma_3\} \models \tau,$$

等等. 如果其中的任何一个满足，那么答案就是“是”. 否则，就要一直检验下去.

由紧致性定理可知，只要 $\Sigma \models \tau$，我们就能够得到肯定的答案. ■

稍后，我们将会使用能行的过程来计算函数. 我们称一个函数是 *能行可计算的*(或简称*可计算的*)，当且仅当对于给定的输入 x，存在能行的过程能够最终得到正确的输出 $f(x)$.

习题

1. 设 Σ 的每个有限子集都是可满足的，证明 $\Sigma; \alpha$ 和 $\Sigma; \neg \alpha$ 中至少有一个也是如此. (这是紧致性定理证明的一部分.)*提示*：如果不是这样，那么对于有限子集 $\Sigma_1 \subseteq \Sigma$ 和 $\Sigma_2 \subseteq \Sigma$, $\Sigma_1; \alpha$ 和 $\Sigma_2; \neg\alpha$ 都是不可满足的. 再考虑 $\Sigma_1 \cup \Sigma_2$.
2. 设 Δ 是满足以下条件的合式公式集合：(i)Δ 的每个有限子集是可满足的，(ii) 对每个合式公式 α，要么 $\alpha \in \Delta$ 要么 $(\neg \alpha) \in \Delta$. 对每个命题符号 A，定义真值指派 v:

$$v(A) = \begin{cases} T & \text{iff} \quad A \in \Delta, \\ F & \text{iff} \quad A \notin \Delta \end{cases}$$

   证明对每个合式公式 φ, $\bar{v}(\varphi) = T$ 当且仅当 $\varphi \in \Delta$. (这也是紧致性定理证明的一部分.)
   提示：对 φ 使用归纳法.
3. 证明由紧致性定理的推论可以证明紧致性定理. (比从头开始证明容易得多.)
4. 1977 年有人证明了每个平面地图都可以用 4 种颜色进行涂色. 当然，地图中只有有限个国家. 我们推广这个概念，设一张无限的平面地图包括无限个国家 $C_1, C_2, C_3, \cdots$. 证明无限平面地图仍可用 4 种颜色涂色. (*提示*：将命题符号分为 4 类，例如，一个命题符号可以用于表达“C_7 涂红色”. 构造合式公式的集合 Σ_1 用以表达，例如，C_7 恰是一种颜色这样的命题. 构造另外一个合式公式的集合 Σ_2，用以表达每一对相邻的国家涂的颜色不同. 然后对 $\Sigma_1 \cup \Sigma_2$ 使用紧致性定理.) 65
5. 若 Σ 是合式公式的集合，定义其上的一个推理为合式公式的有限序列 $\langle \alpha_0, \cdots, \alpha_n \rangle$，对每个 $k \leqslant n$, α_k 至少满足以下 3 条中的一条：(a) α_k 是重言式，(b) $\alpha_k \notin \Sigma$, (c) 对某小于 k 的 i 和 j, α_i 是 $(\alpha_j \to \alpha_k)$. (在 (c) 中，称 α_k 是由 α_i 和 α_j 通过 *假言推理* 获得的.) 请给出如下集合的一个推理，其序列的最终元素是 $\mathbf{P}$.

$$\{\neg \mathbf{S} \vee \mathbf{R}, \mathbf{R} \to \mathbf{P}, \mathbf{S}\}$$

6. 如上题所述，设 $\langle \alpha_0, \cdots, \alpha_n \rangle$ 是合式公式集合 Σ 的一个推理，证明对每个 $k \leqslant n$, $\Sigma \models \alpha_k$. *提示*：对 k 使用归纳法，归纳的假设是：对于每个 $i < k$, $\Sigma \models \alpha_i$.
7. 证明只要 $\Sigma \models \tau$，就存在一个 Σ 的推理，其序列中的最后一个元素是 τ. *说明*：此结果被称为“完备性”；习题 5~7 的概念在 2.4 节中还会出现.

8. 证明定理 17F. 说明：两个半可判定的过程可以构成一个可判定的过程.

*9. 判定性和能行可枚举性的概念不仅可以用在表达式的集合上，还可以用到整数集合或者表达式有序对的集合或者整数有序对的集合中. 证明表达式集合 A 是能行可枚举的，当且仅当存在一个可判定的有序对 $\langle\alpha, n\rangle$(包括一个表达式 α 和一个整数 n) 的集合 B 满足 $A = \operatorname{dom} B$.

10. 设 Σ 是合式公式的能行可枚举集合，假定对每个合式公式 τ，要么 $\Sigma \models \tau$，要么 $\Sigma \models \neg\tau$. 证明 Σ 的重言推论的集合是可判定的.

    (a) 在“互斥或”的情形下证明要么 $\Sigma \models \tau$ 要么 $\Sigma \models \neg\tau$，但二者不能同时成立.

    (b) 在“与或”的情形下证明要么 $\Sigma \models \tau$，要么 $\Sigma \models \neg\tau$，要么二者同时成立.

11. (a) 解释两个能行可枚举集合的并仍是能行可枚举的原因.

    (b) 解释两个能行可枚举集合的交仍是能行可枚举的原因.

12. 对于下列每个条件，给出一个不可满足的公式集合 Γ 的例子，使其满足相应的条件：

    (a) Γ 的每个元素本身是可满足的.

    (b) 对于 Γ 的任意 2 个元素 γ_1 和 γ_2，集合 $\{\gamma_1, \gamma_2\}$ 是可满足的.

    66

    (c) 对于 Γ 的任意 3 个元素 γ_1，γ_2 和 γ_3，集合 $\{\gamma_1, \gamma_2, \gamma_3\}$ 是可满足的.

第2章

一 阶 逻 辑

2.0 预备知识

第 1 章我们介绍了推理思想的第一个数学模型，这个模型非常简单．因而，可以很容易地找到一些直观的、正确推理的例子，而在命题逻辑的模型中无法得到充分反映．

如果有一组 (用自然语言描述的) 条件和一个可能的结论，将其转换到命题逻辑语言中，可以得到条件集合 Σ 和可能的结论 τ．如果 $\Sigma \vDash \tau$，我们就会相信自然语言中的推理是正确的；但是如果 $\Sigma \nvDash \tau$，我们就无法确定了．这说明命题逻辑模型过于简单，还不足以准确地反映自然语言中的推理．

本章将给出一个功能强大的逻辑系统．实际上，可以这么说，本章给出的逻辑系统可以反映任何"勤奋的数学家"对某个问题给出的证明．

首先，我们以非形式的方式来描述一阶逻辑的特征 (至少应该能够模拟)．下面介绍一个特殊的例子，即数论的一阶语言．对这种语言，可以以某种给定的方式 (表 2-1) 与自然语言互相翻译．

表 2-1

形式表达式	给定的翻译
$\mathbf{0}$	"零"．此处 0 是一个常数符号，指定为数字 0 的名字
$\mathbf{S}t$	"t 的后继．" 这里的 S 是一元函数符号，t 是某个数字 a 的表达式．因此，St 指的是 $S(a)$，即 a 的后继．譬如，$S0$ 指的是数字 1
$< v_1 v_2$	"v_1 小于 v_2"．这里的 $<$ 是二元谓词符号．2.1 节的末尾采用惯用的方式，将其表示为一种更常见的方式：$v_1 < v_2$
$\forall$	"对任意自然数．" 这个符号是全称量词符号．一般地，对于每个需要在自然语言中翻译的内容，有一个指定的集合 A(即所谓的论域)．$\forall$ 即指"论域 A 中的每个元素"
$\forall v_1 < \mathbf{0} v_1$	"对每个自然数 v_1，0 小于 v_1"，或者通俗地说，"每个自然数大于 0．" 在给定的翻译中，该形式语言表达的句子是假的，因为 0 不大于其自身

在表 2-1 中，只涉及一个缩写，表 2-2 中给出一些缩写．

表 2-2

缩写表达式	给定的翻译
$x = y$	"x 等于 y"．非缩写形式可变为 $= xy$
$\exists v$	"存在自然数 v 满足" 或者更一般地，"存在全集中的某个元素满足"
$\exists v_1 \forall v_2 v_1 = v_2$	"恰有一个自然数"．同样，在给定的翻译中，该形式句子是错误的
$\forall v_1 (\mathbf{0} < v_1 \vee \mathbf{0} = v_1)$	"每个自然数大于或者等于 $\mathbf{0}$"

事实上，表中不必给出太多，我们可以采用下面两种方法简化，而不会削弱表达能力.

第一种：命题联结符号只用 $\neg$ 和 $\rightarrow$ 两个. 由 1.5 节知道，这已经构成一个完备集合，没有理由使用更多的联结符号.

第二种：不用存在量词 $\exists x$，可以使用 $\neg\forall x\neg$ 来取代. 这样取代没有任何问题，例如自然语言中的句子：

某地有一种东西没有用

等价于

某地不是每样东西都有用

因而公式 $\exists v_1 \forall v_2 v_1 = v_2$，就可以变形为

68

$$(\neg\forall v_1(\neg\forall v_2 = v_1 v_2)).$$

例如，在特定语言中，可以将“张三是人”形式化为 Hs，这里 H 是一元谓词符号，其指定翻译为“是人”；而 s 是一个常数符号，其指定翻译为张三. 类似地，将“张三是好人”译作 Ms. 然后，“全都是好人”可以形式化为 $\forall v_1(Hv_1 \rightarrow Mv_1)$.

在以前的数学课上，读者可能使用过符号 $\forall$ 和 $\exists$. 事实上，在课堂上，有些数学家使用的自然语言经常掺杂一些形式化的符号，我们的一阶语言与之类似就并非偶然了. 我们希望能够设法研究集合论或者群论的命题，而非集合与群本身. (有时我们会使用 *元数学* 这个术语，这个词本身就意味着要从形式化向后退一步，以考察数学家们所做的事情.) 如果我们现在学习逻辑的对象是先前学习集合中所用到的命题，那就需要使用集合论的形式化语言，我们希望形式化语言能够包含集合论的性质.

2.1 一阶语言

假定有无限多个不同的对象 (称之为符号) 排列如下：

A. 逻辑符号

0. 括号：(,).

69

1. 命题联结符号：$\neg$, $\rightarrow$.
2. 变量 (每个对应一个正整数 n)：v_1, v_2, $\cdots$.
3. 等于符号 (可选)：$=$.

B. 参数

0. 量词符号 $\forall$.
1. 谓词符号：对于每一个正整数 n，一 n 元谓词符号集 (可能是空集).
2. 常数符号：一符号集 (可能是空集).
3. 函数符号：对于每一个正整数 n，一 n 元函数符号集 (可能是空集).

在上述 A.3 中的“等于符号”可能会出现，这里我们假定其不出现. 有些语言有“等于符号”，而另外一些语言则没有. “等于符号”是二元谓词符号，其不同于其他二元谓词符号之处在于它是一个逻辑符号而非参数 (这影响到它在自然语言中翻译的方式). 我们确实需要假定每个谓词符号都存在某个 n，来指定为 n 元谓词符号.

在 B.2 中常数符号也称为 0 元函数符号，这样一来，就可以统一处理 B.2 和 B.3 中的符号了.

如前所述，我们假定所有符号都是不同的，并且任何符号都不是其他符号的有限序列组成.

要给定一个语言 (不同于其他一阶语言)，一定要指明两点：(1) 等于符号是否出现；(2) 参数有哪些.

下面给出一些这种语言的例子.

(1) 纯谓词(pure predicate)语言

等于符号：无.

n 元谓词符号：$A_1^n, A_2^n, \cdots$.

常数符号：$a_1, a_2, \cdots$.

n 元函数符号 (n 大于 0)：无.

(2) 集合论语言

等于符号：有 (通常).

谓词参数：一个二元谓词符号 $\in$.

函数符号：无 (或者偶尔用到一个常数符号 $\varnothing$).

(3) 初等(elementary)数论语言(见第 3 章)

等于符号：有.

谓词参数：一个二元谓词符号 $<$.

常数符号：符号 $\mathbf{0}$.

一元函数符号：$\mathbf{S}$(后继).

二元函数符号：$+$(加法)，$\cdot$(乘法) 和 $\mathbf{E}$(指数).

70

在例 2 和例 3 中，参数具有特定的翻译. 我们会分别给出两类命题的例子：(1) 可以用这些语言翻译的；(2) 不能用这些语言翻译的.

注意：我们的语言的概念包含集合论语言，这一点非常重要. 一个非常普遍的观点是数学可以嵌入到集合论中. 这就意味着

(a) 数学中的命题 (比如算术基本定理) 都可以用集合论语言进行表述；

(b) 数学中的定理可以使用集合论中的公理逻辑地推导出来.

一阶逻辑的模型可以完整而充分地反映这一点.

集合论语言的例子 此处 $\forall$ 的意思是“对所有集合”，而 $\in$ 则意味着“是……的一个元素”.

(1) “不存在一个集合使每个集合都是它的元素.” 几步就可以将此命题用集合论语言形式化. 在这一过程中的命题既非自然语言中的命题，也不是形式语言中的命题，而是混合语言的.

$$\neg\ [\text{存在一个集合使每个集合都是它的元素}]$$

$$\neg\ \exists v_1[\text{使每个集合都是}v_1\text{的元素}]$$

$$\neg\ \exists v_1 \forall v_2 v_2 \in v_1$$

由于谓词符号总是出现在上下文的最左边，因此，我们需要将 $v_2 \in v_1$ 替换为 $\in v_2 v_1$. 另外，要用 $\neg \forall v_1 \neg$ 替换 $\exists v_1$，且需要使用适当数量的括号. 最后的结果应该是

$$(\neg\ (\neg\ \forall v_1(\neg\ \forall v_2 \in v_2 v_1))).$$

(2) 序对公理："对任意两个集合，都存在一个集合只以这两个集合作它的元素．" 同样用几步来完成转换．

$$\forall v_1 \forall v_2[\text{都存在一个集合只以这两个集合作它的元素}]$$
$$\forall v_1 \forall v_2 \exists v_3[\text{只有 } v_1, v_2 \text{ 是 } v_3 \text{的元素}]$$
$$\forall v_1 \forall v_2 \exists v_3 \forall v_4(v_4 \in v_3 \leftrightarrow v_4 = v_1 \vee v_4 = v_2)$$

现在用 $\neg \forall v_3 \neg$ 替换 $\exists v_3$，用 $\in v_4 v_3$ 替换 $v_4 \in v_3$，并且用 $= v_4 v_i$ 替换 $v_4 = v_i$．此外，使用 $\rightarrow$ 和 $\neg$ 消除 $\leftrightarrow$ 和 $\wedge$．这样

$$\alpha \vee \beta \text{ 变成 } \neg \alpha \rightarrow \beta;$$
$$\alpha \leftrightarrow \beta \text{ 变成 } \neg((\alpha \rightarrow \beta) \rightarrow \neg(\beta \rightarrow \alpha)).$$

71

最终结果就是：

$$\forall v_1 \forall v_2(\neg \forall v_3(\neg \forall v_4(\neg((\in v_4 v_3 \rightarrow ((\neg = v_4 v_1) \rightarrow$$
$$= v_4 v_2)) \rightarrow (\neg(((\neg = v_4 v_1) \rightarrow = v_4 v_2) \rightarrow \in v_4 v_3)))))).$$

这个结果的易读性不如前面的结果，因而，对这样的结果我们并不感兴趣．最终，我们会采用一种方便的方式来避免出现这样的结果．尽管如此，现在我们还是要给出这样的结果．

初等数论语言的例子　此处 $\forall$ 的意思是"对所有的自然数"，而 $<, \mathbf{0}, \mathbf{S}, +, *, \mathbf{E}$ 的意思是显而易见的．

(1) 由于自然数 2 是 0 的后继的后继，因此，可以使用 **SS0** 来表示；类似地，对于 4 则表示为 **SSSS0**．表达式"2+2"可以表示为 **SS0** + **SS0**，可是，因为函数符号总是出现在最左边 (也就是采用函数的波兰记法)，因此，对应的"2+2"应该使用 +**SS0SS0** 来表示．自然语言句子"二加二等于四．"要转换为

$$= +\mathbf{SS0}\ \mathbf{SS0}\ \mathbf{SSSS0}.$$

(空格可以用于帮助阅读，但并非该语言的正式组成部分．)

(2) 下面用 3 步将"任意非零自然数都是某个自然数的后继"转换成形式语言．

$$\forall v_1[\text{如果 } v_1 \text{ 非零，则 } v_1 \text{ 是某个自然数的后继}].$$
$$\forall v_1(v_1 \neq \mathbf{0} \rightarrow \exists v_2 v_1 = \mathbf{S} v_2).$$
$$\forall v_1((\neg = v_1 \mathbf{0}) \rightarrow (\neg \forall v_2(\neg = v_1 \mathbf{S} v_2))).$$

(3) "任意非空的自然数集都有最小数．"这个命题不能用初等数论语言来形式化，原因在于我们无法表达"任意集合"．转换这个命题，要求使用集合论的 (一阶) 语言，或者数论的二阶语言．然而，"素数集合有最小元．"是可以转换的．(第一步将此命题转换为"存在最小素数"，其他步骤请读者自行完成，在本节中会有相关的提示．)

特定语言的例子

(1) "所有苹果都是坏的．"

72

$$\forall v_1(A v_1 \rightarrow B v_1).$$

(2) “有些苹果是坏的.”

中间步骤：$\exists v_1(Av_1 \land Bv_1)$.

最后结果：$(\neg\, \forall v_1(\neg\, (\neg\, (Av_1 \to (\neg\, Bv_1)))))$.

这两个例子描述了两种频繁出现的模式. 在自然语言中，用于声明某个类中的每一个对象都具有某个属性的命题，可以转换为

$$\forall v(_ \to _).$$

用于声明某个类中的某些对象具有某个属性的命题可以译作：

$$\exists v(_ \land _).$$

请读者不要混淆这两种模式. 例如，

$$\forall v_1(Av_1 \land Bv_1)$$

翻译为“每个东西都是苹果，并且都是坏的.”比第一个例子中的断言要强一些. 类似地，$\exists v_1(Av_1 \to Bv_1)$，翻译为“有些东西是坏的，只要它是苹果.”就比第二个例子中的意思要弱一些. 如果世界上没有苹果，即使所有的苹果都是好的，这个说法也是对的.

(3) “博比的父亲可以把这一地段任何一个孩子的父亲打败.”构造一种语言，其中 $\forall$ 表示“对所有人”，Kx 表示“x 是这一地段的一个孩子”，b 表示“博比”，Bxy 表示“x 可以打败 y”，且 fx 表示“x 的父亲.”，那么该句子就可以变成：

$$\forall v_1(Kv_1 \to ((\neg\ = v_1 b) \to Bfbfv_1)).$$

(4) 在微积分中，我们学习过“在 x 接近 a 时，函数 f 接近 L”：

$$\forall \varepsilon(\varepsilon > 0 \to \exists \delta(\delta > 0 \land \forall x(|x-a| < \delta \to |fx - L| < \varepsilon))).$$

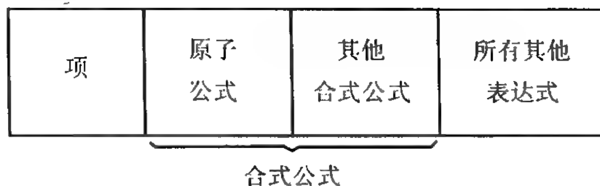
如果不考虑记法，对这个公式，我们感兴趣的是它用到了排序符号、函数符号 f、减号、绝对值符号等谓词符号以及 $\mathbf{0}$、a 和 L 等常数符号.

2.1.1 公式

表达式 是符号的任意有限序列. 大多数表达式是没有意义的，但是项和合式公式是具有特定意义的表达式.

73

项是指语言中的名词和代词，是可以翻译成对象名称的表达式. 原子公式是指那些没有使用联结符号和量词符号的合式公式.



项定义为在常数符号和变量之前加上函数符号构成的表达式. 为了使用第 1 章中的术语重述这一点，每个 n 元函数符号 f 得到如下形式的一个 n 元项构造运算 $\mathcal{F}_f$：

$$\mathcal{F}_f(\varepsilon_1, \cdots, \varepsilon_n) = f\varepsilon_1 \cdots \varepsilon_n.$$

定义　项 的集合是指由常数符号和变量通过使用 0 次和多次运算 $\mathcal{F}_f$ 得到的表达式的集合.

如果没有函数符号 (常数符号除外)，那么项恰好就是常数符号和变量. 在这种情况下，我们无需归纳定义.

注意：对于项，可以使用波兰记法，把函数符号放在左边. 项不包括括号和逗号. 稍后，我们将证明它的唯一可翻译性：对给定的项，可以毫无二义地将其分解.

项是可以翻译成对象名称 (名词短语) 的表达式，而合式公式则可以翻译成关于对象的断言.

以下是数论语言中的一些相关的例子：

$$+v_2\mathbf{S0},$$
$$\mathbf{SSSS0},$$
$$+\mathbf{E}v_1\mathbf{SS0E}v_2\mathbf{SS0}.$$

原子公式的作用相当于命题逻辑中的命题符号，原子公式是具有如下形式的表达式：

$$Pt_1 \cdots t_n,$$

其中 P 是 n 元谓词符号，$t_1, \cdots, t_n$ 是项.

例如，$= v_1v_2$ 是一个原子公式，因为等号是一个二元谓词符号，并且每个变量都是项. 在集合论语言中 $\in v_5v_3$ 是原子公式.

74

注意：原子公式不是归纳定义的，相反地，我们只是简单而明白地说哪些表达式是原子公式. 合式公式是指由原子公式通过使用 0 次或多次联结符号和量词符号构成的表达式. 使用第 1 章中的术语，通过定义表达式上的公式构造运算，可以重述如下：

$$\mathcal{E}_\neg(\gamma) = (\neg\ \gamma),$$

$$\mathcal{E}_\neg(\gamma, \delta) = (\gamma \to \delta),$$

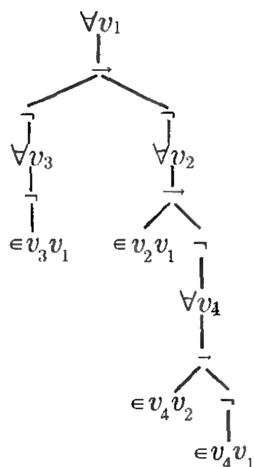
$$\mathcal{Q}_i(\gamma) = \forall v_i \gamma.$$

定义　合式公式 (或公式) 的集合是指由原子公式通过 0 次或多次使用 $\mathcal{E}_\neg, \mathcal{E}_\to$ 和 $\mathcal{Q}_i(i = 1, 2, \cdots)$ 运算构成的表达式的集合.

例如，一方面，$\neg\ v_3$ 不是一个合式公式. (为什么？) 而另一方面，

$$\forall v_1((\neg\ \forall v_3(\neg\ \in v_3v_1)) \to (\neg\ \forall v_2(\in v_2v_1 \to (\neg \forall\ v_4(\in v_4v_2 \to (\neg\ \in v_4v_1))))))$$

是一个合式公式，如下图中的树所示：



75

进一步学习，会发现这个合式公式是集合论中的“正则公理”.

2.1.2 自由变量

$\forall v_2 \in v_2 v_1$ 和 $(\neg\forall\ v_1(\neg\forall v_2 \in v_2 v_1))$ 是两个合式公式的例子，但这两个例子有很大的差别. 第 2 个例子在自然语言中的翻译可能是：

存在这样一个集合，每个集合都是其元素.

第 1 个例子，在自然语言中的翻译只能是一个不完整的句子：

每个集合是 ___ 的一个元素.

因为不知道 v_1 代表什么，所以无法完成这个句子. 在这种情况下，称 v_1 在合式公式 $\forall v_2 \in v_2 v_1$ 中 **自由出现**. 当然，我们需要一个精确的定义，这个定义仅仅是关于符号本身的，而与在自然语言中可能的翻译无关.

考虑任意变量 x，对每一个合式公式 α，对“x 在 α 中自由出现”的递归定义如下：

(1) 对原子公式 α，x 在 α 中自由出现当且仅当 x 出现在 α 中 (即作为一个符号出现).

(2) x 在 $(\neg\ \alpha)$ 中自由出现当且仅当 x 在 α 中自由出现.

(3) x 在 $(\alpha \to \beta)$ 中自由出现当且仅当 x 在 α 中或者在 β 中自由出现.

(4) x 在 $\forall v_i \alpha$ 中自由出现当且仅当 x 在 α 中自由出现且 $x \neq v_i$.

该定义隐含使用了递归定理，我们可以使用函数重述这个定义. 首先从定义在原子公式上的函数 h 开始：

$$h(\alpha) = \text{所有变量的集合，如果有的话，也包括原子公式 } \alpha$$

进一步扩展 h 到定义在所有合式公式上的函数 $\overline{h}$：

$$\begin{aligned}\overline{h}(\mathcal{E}_{\neg}(\alpha)) &= \overline{h}(\alpha),\\ \overline{h}(\mathcal{E}_{\to}(\alpha,\beta)) &= \overline{h}(\alpha) \cup \overline{h}(\beta),\\ \overline{h}(\mathcal{Q}_i(\alpha)) &= \overline{h}(\alpha) \text{ 如果可以，除去 } v_i \text{ 后}\end{aligned}$$

这样，我们就说 x 在 α 中自由出现 (或者说 x 是 α 的一个 **自由变量**) 当且仅当 $x \in \overline{h}$. 由 1.4 节中的递归定理和公式的唯一可分解性 (在 2.3 节证明) 可知，这样的 $\overline{h}$ 是唯一存在的.

76

如果合式公式 α 中没有自由变量出现 (即 $\overline{h}(\alpha) = \varnothing$), 那么 α 就是一个*句子*. (只要给出参数的含义, 句子直觉地看是不用括号就可以用自然语言翻译的合式公式.)

例如, $\forall v_2(Av_2 \to Bv_2)$ 和 $\forall v_3(Pv_3 \to \forall v_3 Qv_3)$ 都是句子, 但在 $(\forall v_1 Av_1 \to Bv_1)$ 中 v_1 是自由出现的. 句子是最有意义的合式公式, 其余的合式公式则次之, 它们是用于构造句子的基本模块.

在翻译自然语言中的某个句子时, 无需选择特定的变量. 前面, 我们将 "所有苹果都是坏的" 译作 "$\forall v_1(Av_1 \to Bv_1)$", 实际上译作 $\forall v_{27}(Av_{27} \to Bv_{27})$ 也是完全一样的.

正如在自然语言中说 "对于任意对象, 如果它是一个苹果, 那么它就是坏的" 中的 "它 (it)" 一样, 这里的变量实际上是一个代词. (在我们的语言中有许多代词: $\text{it}_1, \text{it}_2, \cdots$), 由于变量选择是不重要的, 也就无需指定特定的变量. 例如, $\forall x(Ax \to Bx)$, 显而易见, x 是变量. (变量选择的无关性是一个定理.)

类似变量的使用方法在数学中也经常出现, 在

$$\sum_{i=1}^{7} a_{ij}$$

中 i 是一个*假变量*(dummy variable), 而 j 是自由出现的.

2.1.3　符号

可以通过显式书写每一个符号的方式来指定一个合式公式 (或者任意的表达式), 例如

$$\forall v_1((\neg = v_1 0) \to (\neg \forall v_2(\neg = v_1 \mathbf{S} v_2))).$$

尽管这种方式书写完整, 但是却不易理解. 这种不易理解有损于我们对语言简化的要求 (比如不使用存在量词符号). 因此我们宁愿选择使用非直接、却易读的方式来表达合式公式. 这样, 我们就可以使用类似

$$\forall v_1(v_1 \neq \mathbf{0} \to \exists v_2 v_1 = \mathbf{S} v_2)$$

这样的方式来表示前面出现的合式公式.

注意, 我们没有改变合式公式的定义, 而只是简单地采用某种特定的方式来描述合式公式. 在符号顺序非常重要的情况下 (这种情况很少出现), 不能使用这种方便的表示方法, 而必须使用最基本的记法.

77

下面是一些缩写和惯用的记法. 这里的 α 和 β 是公式, 而 x 是变量, u 和 t 是项.

$(\alpha \vee \beta)$ 简化 $((\neg\, \alpha) \to \beta)$.

$(\alpha \wedge \beta)$ 简化 $(\neg\, (\alpha \to (\neg\, \beta)))$.

$(\alpha \leftrightarrow \beta)$ 简化 $((\alpha \to \beta) \wedge (\beta \to \alpha))$; 即 $(\neg\, ((\alpha \to \beta) \to (\neg\, (\beta \to \alpha))))$.

$\exists x \alpha$ 简化 $(\neg\, \forall x(\neg\, \alpha))$.

$u = t$ 代替 $= ut$, 类似地缩写可以用于其他的二元谓词和函数符号. 例如, $2 < 3$ 是 < 23 的缩写, $2 + 2$ 是 $+22$ 的缩写.

$u \neq t$ 是 $(\neg\, = ut)$ 的缩写, 类似地, $u \nless t$ 是 $(\neg\, < ut)$ 的缩写.

括号可以使用 (,) 和 [,] 等. 我们会尽可能地省略它们, 这样, 我们采用的惯用记法如下:

(1) 最外层的括号可以省略．如 $\forall x\alpha \to \beta$ 即 $(\forall x\alpha \to \beta)$；

(2) 尽量少地使用 $\neg$, $\forall$ 和 $\exists$. 如，

$$\neg\,\alpha \wedge \beta \text{ 是 } ((\neg\,\alpha) \wedge \beta)\text{，而不是 } \neg\,(\alpha \wedge \beta)\text{；}$$
$$\forall x\alpha \to \beta \text{ 是 } (\forall x\alpha \to \beta)\text{，而不是 } \forall x(\alpha \to \beta)\text{；}$$
$$\exists x\alpha \wedge \beta \text{ 是 } (\exists x\alpha \wedge \beta)\text{，而不是 } \exists x(\alpha \wedge \beta).$$

在这些情况下，可以添加括号，如 $(\exists x\alpha) \wedge \beta$.

(3) 与第 2 条类似，尽量少使用 $\wedge$ 和 $\vee$. 如

$$\neg\,\alpha \wedge \beta \to \gamma \text{ 是 } ((\neg\,\alpha) \wedge \beta) \to \gamma.$$

(4) 当联结符号重复出现时，表达式从右边开始分组．例如，

$$\alpha \to \beta \to \gamma \text{ 是 } \alpha \to (\beta \to \gamma).$$

例 将缩写形式的公式改写为每个符号都按顺序显式出现的非缩写形式：

(1) $\exists x(Ax \wedge Bx)$ 是 $(\neg\,\forall x(\neg\,(\neg\,(Ax \to (\neg\,Bx)))))$. $(\neg\,\forall x(Ax \to (\neg\,Bx)))$ 也是一个与之等价的公式 (无论从何种意义上说都是等价的).

(2) $\exists xAx \to Bx$ 是 $((\neg\,\forall x(\neg\,Ax)) \to Bx)$.

$\exists x(Ax \to Bx)$ 是 $(\neg\,\forall x(\neg\,(Ax \to Bx)))$.

除特殊情况外，我们将会尽量使用下面的字母表． 78

- 谓词符号：大写斜体的字母和 $\in$, $<$.
- 变量：v_i, u, v, x, y, z.
- 函数符号：f, g, h 和 $\mathbf{S}$, $+$ 等.
- 常数符号：$a, b, \cdots$, 和 $\mathbf{0}$.
- 项：u, t.
- 公式：小写希腊字母.
- 句子：σ 和 τ.
- 公式集：大写希腊字母和特定的斜体字母 (看作是希腊字母) 即 A(alpha) 和 T(tau).
- 结构 (见 2.2 节)：大写德文字母.

习题

1. 假定某种语言的参数如下，$\forall$：含义为“对所有事物”；N：含义为“是一个数字”；I：含义为“有意义”；$<$：含义为“小于”；$\mathbf{0}$ 是一个常数符号表示零．试将如下列出的自然语言句子翻译到这种语言中，若自然语言句子的含义是模糊的，则要给出多种翻译.
   (a) 0 小于任意数字.
   (b) 如果任意数字是有意义的，那么零是有意义的.
   (c) 没有小于零的数字.
   (d) 任意没有意义的数字具有如下性质，如果比它小的所有数字有意义，那么该数字也有意义.
   (e) 没有一个数字使得所有的数字都小于它.
   (f) 没有一个数字使得没有数字比它小.
2. 对于上题中的语言，将如下合式公式转译回自然语言中去：

$$\forall x(Nx \to Ix \to \neg \forall y(Ny \to Iy \to \neg x < y)).$$

将习题 3~8 中的每个自然语言句子翻译为指定的一阶语言. (可以参考例题中的步骤完成.) 充分使用惯用记法和缩写以使得最终结果尽可能具有易读性.

3. a 与 b 都不是所有集合的元素. ($\forall$: 对所有集合; $\in$: 是……的一个元素; a: a; b: b.)
4. 如果马是动物, 那么马的头是动物的头. ($\forall$: 对所有事物; E: 是一匹马; A: 是一个动物; hx: x 的头.)
5. (a) 任何时间你都可以欺骗某些人. (b) 某些时间你可以欺骗所有的人. (c) 你不能在任何时间欺骗所有的人. ($\forall$: 对所有事物; P: 是一个人; T: 是一个时间; Fxy: 你可以在 y 时间欺骗 x. 上述句子中可能有一个或者多个是含糊的, 需要给出多种翻译.)

79

6. (a) Adams 不能正确完成每一项任务. (b)Adams 不能正确完成任何任务. ($\forall$: 对所有事物; J: 是一个任务; a: Adams; Dxy: x 能够正确完成 y.)
7. (a) 没有人喜欢每一个人. (b) 没有民主党人喜欢每一个共和党人. ($\forall$: 对所有人; D: 是一个民主党人; R: 是一个共和党人; Lxy: x 喜欢 y.)
8. (a) 每一个有驴子的农民都需要干草. (b) 每一个有驴子的农民都打驴子. ($\forall$: 对所有事物; F: 是一个农民; D: 是一头驴子; Oxy: x 拥有 y; H: 需要干草; Bxy: x 打 y.)
9. 给出一个精确的定义说明: 变量 x 在合式公式 α 中作为第 i 个符号自由出现. (如果 x 是第 i 个符号但不是自由出现的, 那么称它 *约束* 出现在这个位置上.)
10. 以按照实际顺序显式列出每一个符号的方式重写下列每个合式公式:

    (a) $\exists v_1 Pv_1 \wedge Pv_1$

    (b) $\forall v_1 Av_1 \wedge Bv_1 \to \exists v_2 \neg Cv_2 \vee Dv_2$

    说出每个合式公式中的自由变量.

2.2 真值与模型

在命题逻辑中, 用真值指派说明命题符号的真与假. 在一阶逻辑中则使用 *结构* (structure), 结构可以看作是为把形式语言翻译到自然语言而提供的字典. (结构有时候也称为 *解释* (interpretation), 但是我们将会把解释一词用于另外一个概念, 见 2.7 节).

一阶语言的结构要指明:

(1) 全称量词 ($\forall$) 所指的事物的集合是什么,

(2) 其他参数 (谓词和函数符号) 的含义.

80

形式上, 一阶语言的一个 *结构* $\mathfrak{A}$ 是一个函数, 其定义域为参数的集合, 且满足 [1]

(1) $\mathfrak{A}$ 为全称量词 $\forall$ 指派一个非空集合 $|\mathfrak{A}|$, 称为 $\mathfrak{A}$ 的 *论域* (universe) (或者 *定义域* (domain)).

(2) $\mathfrak{A}$ 给每一个 n 元谓词符号 P 指派一个 n 元关系, $P^{\mathfrak{A}} \subseteq |\mathfrak{A}|^n$, 即 $P^{\mathfrak{A}}$ 是 P 上一个 n 元组的集合.

(3) $\mathfrak{A}$ 给每个常数符号 c 指派一个论域 $|\mathfrak{A}|$ 中的元素 $c^{\mathfrak{A}}$.

(4) $\mathfrak{A}$ 给每个 n 元函数符号 f 指派一个 $|\mathfrak{A}|$ 上的 n 元运算 $f^{\mathfrak{A}}$; 即 $f^{\mathfrak{A}} : |\mathfrak{A}|^n \to |\mathfrak{A}|$.

基本思想就是 $\mathfrak{A}$ 给参数赋以意义. $\forall$ 即是"对 $|\mathfrak{A}|$ 中的所有事物"; 符号 c 则是给 $c^{\mathfrak{A}}$ 命名; 原子公式 $Pt_1 \cdots t_n$ 是以 $t_1, \cdots, t_n$ 命名的在关系 $P^{\mathfrak{A}}$ 中的 n 元组. (稍后, 我们将详细地重申这些条件.)

1. 符号 $\mathfrak{A}$ 是德文字母表中的字母 A, 其后面的两个字母是 $\mathfrak{B}$ 和 $\mathfrak{C}$.

注意到 $|\mathfrak{A}|$ 必须是非空的. 同时 $f^{\mathfrak{A}}$ 的定义域必须包含全体 $|\mathfrak{A}|^n$. 目前, 我们还不涉及部分定义函数.

例 考虑集合论语言, 其唯一的参数是 $\in$(除 $\forall$ 外). 给定结构 $\mathfrak{A}$ 如下:

$$|\mathfrak{A}| = \text{自然数集},$$
$$\in^{\mathfrak{A}} = \text{使得 } m < n \text{ 的所有序对 } \langle m, n\rangle \text{ 组成的集合}.$$

(这样, 可以将 $\in$ 译作 "小于".) 使用一个结构, 我们可以将形式语言中的句子翻译到自然语言中去, 并能够试图解释翻译的真假. 一阶语言句子

$$\exists x \forall y \neg\, y \in x$$

(或者更规范一些, $(\neg\forall v_1(\neg\forall v_2(\neg \in v_2 v_1))))$, 在 $\mathfrak{A}$ 的解释下可以译作一个真的句子:

存在一个自然数, 没有比它更小的自然数.

而在集合论的翻译中断定为存在空集. 这样, 我们称在 $\mathfrak{A}$ 中的 $\exists x\forall y \neg y \in x$ 是*真的*, 或者 $\mathfrak{A}$ 是这个句子的*模型*. 另一方面, $\mathfrak{A}$ 不是序对公理

$$\forall x \forall y \exists z \forall t (t \in z \leftrightarrow t = x \vee t = y)$$

的模型, 其原因在于该句子在 $\mathfrak{A}$ 中的翻译是假的, 没有自然数 m 能够使得对每个 n,

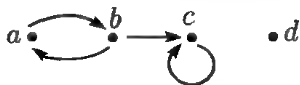
$$n < m \quad \text{iff} \quad n = 1.$$

(熟悉公理集合论的读者可以验证 $\mathfrak{A}$ 是外延公理, 并集公理和正则公理的模型.) 81

例 假定语言只含有一个参数 $\forall$ 和一个二元谓词符号 E. 现在考虑*有限*结构 $\mathfrak{B}$, 其论域 $|\mathfrak{B}|$ 包含 4 个不同的对象 $\{a, b, c, d\}$. 假定二元关系 $E^{\mathfrak{B}}$ 是如下对偶集合:

$$E^{\mathfrak{B}} = \{\langle a, b\rangle, \langle b, a\rangle, \langle b, c\rangle, \langle c, c\rangle\}.$$

则 $\mathfrak{B}$ 的*有向图* (directed graph) 如下, 其顶点集为论域 $\{a, b, c, d\}$.



这里 Exy 表示从顶点 x 到顶点 y 的边. (如果二元关系 $E^{\mathfrak{B}}$ 是对称的, 那么可以用无向图来表示.)

在结构 $\mathfrak{B}$ 的解释下, 句子 $\exists x \forall y \neg\, y E x$ 可以翻译为 "存在一个顶点使得每个顶点都没有边通向该顶点". (这里的自然语言比符号化的语言更难读懂 !) 这个句子在 $\mathfrak{B}$ 中是真的, 因为的确是存在一个顶点 d, 没有边指向 d.

从前面的例子中, 可以直观又清楚地看到: 形式语言的句子在某些结构下是真的, 而在另外一些结构下则是假的. 现在我们需要对 "σ 在 $\mathfrak{A}$ 中是真的" 做简洁的数学定义. 定义要使用数学术语, 而无需翻译到自然语言, 也无需某种假定的标准来说明句子的真假. (如果需

要这样的标准，可以试一下这个句子“这个句子是假的”．）换句话说，就是要给“σ 在 $\mathfrak{A}$ 中是真的”一个非形式的概念，使之成为 *数学* 的一部分．

为了定义“σ 在 $\mathfrak{A}$ 中是真的”，即对句子 σ 和结构 $\mathfrak{A}$，有 $\models \mathfrak{A}\sigma$．需要首先给出包括合式公式的更为一般的概念．令

- φ 是语言中的合式公式．
- $\mathfrak{A}$ 是语言的结构．(82)
- $s: V \to |\mathfrak{A}|$ 是从集合 V 中的所有变量到 $\mathfrak{A}$ 的论域 $|\mathfrak{A}|$ 的函数．

那么定义对于 $\mathfrak{A}$，s *满足* φ 的含义为

$$\models_{\mathfrak{A}} \varphi[s].$$

非形式地说，$\models_{\mathfrak{A}} \varphi[s]$ 当且仅当由 $\mathfrak{A}$ 确定的 φ 的翻译是真的，这里的变量 x 无论其是否自由出现都翻译为 $s(x)$．其形式定义如下：

I. *项*．定义扩充

$$\bar{s} : T \to |\mathfrak{A}|,$$

为一个从所有项的集合 T 到函数 $\mathfrak{A}$ 的论域的函数．其思想是：$\bar{s}(t)$ 应该是 $|\mathfrak{A}|$ 中的由 t 命名的元素．$\bar{s}$ 的递归定义如下：

(1) 对每一个变量 x，$\bar{s}(x) = s(x)$．

(2) 对每一个常数符号 c，$\bar{s}(c) = c^{\mathfrak{A}}$．

(3) 如果 $t_1, \cdots, t_n$ 是项，且 f 是一个 n 元函数符号，那么

$$\bar{s}(ft_1 \cdots t_n) = f^{\mathfrak{A}}(\bar{s}(t_1), \cdots, \bar{s}(t_n)).$$

对 $n = 1$ 的交换图 (commutative diagram) 如下：

$$\begin{array}{ccc} T & \xrightarrow{\bar{s}} & |\mathfrak{A}| \\ \mathscr{F}_f \downarrow & & \downarrow f^{\mathfrak{A}} \\ T & \xrightarrow{\bar{s}} & |\mathfrak{A}| \end{array}$$

$\bar{s}$ 扩充 s 的唯一存在性可以由递归定理 (1.4 节)，通过项具有唯一可分解性导出 (2.3 节)．注意 $\bar{s}$ 依赖于 s 和 $\mathfrak{A}$．(事实上，对于 $\bar{s}(t)$ 的一个合理的记法应该是 $t^{\mathfrak{A}}[s]$，这样能显式地说明对 $\mathfrak{A}$ 的依赖．)

II. *原子公式*．

原子公式可以不用归纳而显式地定义．因此，原子公式的 *满足*(satisfaction) 的定义也是不使用递归的直接定义．

(1) $\models_{\mathfrak{A}} = t_1t_2[s]$　iff　$\bar{s}(t_1) = \bar{s}(t_2)$．(这里的 $=$ 是一个逻辑符号，不是需要解释的参数．)

(83) (2) 对 n 元谓词参数 P，

$$\models_{\mathfrak{A}} Pt_1 \cdots t_n[s] \quad \text{iff} \quad \langle \bar{s}(t_1), \cdots, \bar{s}(t_n) \rangle \in P^{\mathfrak{A}}.$$

III. *其他合式公式*．合式公式是归纳定义的，因而其满足的定义也是递归的．

(1) 对原子公式的定义如上．

(2) $\models_{\mathfrak{A}} \neg\varphi[s]$ 当且仅当 $\not\models_{\mathfrak{A}} \varphi[s]$.

(3) $\models_{\mathfrak{A}} (\varphi \to \psi)[s]$ 当且仅当或者 $\not\models_{\mathfrak{A}} \varphi[s]$ 或者 $\models_{\mathfrak{A}} \psi[s]$ 或者二者都成立. (换句话说，如果 $\mathfrak{A}$ 以 S 满足 φ，那么 $\mathfrak{A}$ 以 S 满足 ψ.)

(4) $\models_{\mathfrak{A}} \forall x\varphi[s]$ 当且仅当对每个 $d \in |\mathfrak{A}|$，有 $\models_{\mathfrak{A}} \varphi[s(x|d)]$.

这里的 $s(x|d)$ 是一个函数，其取值除了在变量 x 点取值为 d 外，其他的取值与函数 s 相同. 即

$$s(x|d)(y) = \begin{cases} s(y) & \text{如果} \quad y \neq x, \\ d & \text{如果} \quad y = x. \end{cases}$$

(这里 $\forall$ 的含义是“对 $|\mathfrak{A}|$ 中的所有对象”.) 至此，读者可能需要重新考虑一下前面关于 $\models_{\mathfrak{A}} \varphi[s]$ 的非形式化定义，并观察它是如何形式化的.

应该着重说明的是，关于满足的定义是递归定理的另外一个应用，并使用了合式公式的唯一分解性. 定义可以从函数的角度重述，这样可以清楚地看出 1.4 节的递归定理是如何应用的：

(i) 给定 $\mathfrak{A}$.

(ii) 定义函数 $\bar{h}$(对定义在原子公式上的函数的 h 的扩充) 满足：对任给的合式公式 φ，$\bar{h}(\varphi)$ 是从 V 到 $|\mathfrak{A}|$ 的函数集.

(iii) 定义

$$\models_{\mathfrak{A}} \varphi[s] \quad \text{iff} \quad s \in \bar{h}(\varphi).$$

至于函数 h 的明确定义及其对扩充 $\bar{h}$ 的唯一确定性，我们留给读者做练习 (习题 7). 另外一个完美的定义方式是令 $\bar{h}(\varphi)$ 是这样的函数集：其定义域为在 φ 中自由出现的变量 x 的集合.

例 假定语言中有参数 $\forall$, P(二元谓词符号), f(一元函数符号) 和 c(常数符号). 设该语言的结构 $\mathfrak{A}$ 定义如下：

$|\mathfrak{A}| = \mathbb{N}$, 自然数集,

$P^{\mathfrak{A}} =$ 使得 $m \leqslant n$ 的序对 $\langle m, n\rangle$ 的集合,

$f^{\mathfrak{A}} =$ 后继函数 S；$f^{\mathfrak{A}}(n) = n + 1$,

$c^{\mathfrak{A}} = 0$.

事实上 $\mathfrak{A}$ 是一个函数，由此可以将上述总结为一行：

84

$$\mathfrak{A} = (\mathbb{N}; \leqslant, S, 0).$$

只有当上下文能够确定要素与参数的对应关系后，上述记法才是明确的.

令 $s : V \to \mathbb{N}$ 是这样的函数：$s(v_i) = i - 1$；即 $s(v_1) = 0$，$s(v_2) = 1$，依此类推.

(1) $\bar{s}(ffv_3) = S(S(2)) = 4$ 且 $\bar{s}(fv_1) = S(0) = 1$.

(2) $\bar{s}(c) = 0$ 且 $\bar{s}(ffc) = 2$；不使用 s.

(3) $\models_{\mathfrak{A}} Pcfv_1[s]$. 当然，这是非形式化的，因为将其翻译回到自然语言时，会得到真命题“$0 \leqslant 1$”. 更规范地说，其原因在于

$$\langle \bar{s}(c), \bar{s}(fv_1)\rangle = \langle 0, 1\rangle \in P^{\mathfrak{A}}.$$

(4) $\models_{\mathfrak{A}} \forall v_1 Pcv_1$. 翻译为自然语言是"0 小于等于任意自然数". 这需要对 $\mathbb{N}$ 中的每个 n 进行验证：

$$\models_{\mathfrak{A}} Pcv_1[s(v_1|n)],$$

可以简化为

$$\langle 0, n\rangle \in P^{\mathfrak{A}},$$

即 $0 \leqslant n$.

(5) $\not\models_{\mathfrak{A}} \forall v_1 Pv_2v_1[s]$, 因为存在自然数 m 满足

$$\not\models_{\mathfrak{A}} Pv_2v_1[s(v_1|m)];$$

即

$$\langle s(v_2), m\rangle \notin P^{\mathfrak{A}}.$$

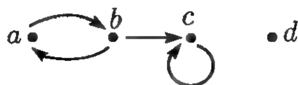
事实上，由于 $s(v_2) = 1$，我们必须取 m 为 0.

注意：不要混淆相关的记法，如函数 符号f 和 函数$f^{\mathfrak{A}}$.

例　前面考察过关于如下结构 $\mathfrak{B}$ 的例子：

$$|\mathfrak{B}| = \{a, b, c, d\}, E^{\mathfrak{B}} = \{\langle a, b\rangle\langle b, a\rangle, \langle b, c\rangle\langle c, c\rangle\},$$

85 其中语言参数为 $\forall$ 和 E：



那么 $\models_{\mathfrak{B}} \forall v_2 \neg Ev_2v_1[s]$　iff　$s(v_1) = d$, 即，没有指向顶点 d 的边，但是这样的顶点仅 d 一个. 该公式的否定为 $\models_{\mathfrak{B}} \exists v_2 Ev_2v_1[s]$　iff　$s(v_1) \in \{a, b, c\}$.

当我们想知道结构 $\mathfrak{A}$ 是否以 s 满足合式公式 φ 时，无需使用 s 提供的所有 (数量无限多的) 信息，这一点暂时先不验证. 问题的关键在于 s 在 φ 中的自由变量 (有限多的) 上的取值. 特别地，如果 φ 是一个句子，那么 s 就无关紧要了.

定理 22A　假定 s_1 和 s_2 是从 V 到 $|\mathfrak{A}|$ 中的函数，它们在合式公式 φ 中自由出现的所有变量上的取值相同，那么

$$\models_{\mathfrak{A}} \varphi[s_1] \quad \text{iff} \quad \models_{\mathfrak{A}} \varphi[s_2].$$

证明　因满足是递归定义的，所以证明使用归纳法. 对给定的结构 $\mathfrak{A}$，使用归纳法证明每个合式公式 φ 具备如下特性：只要函数 s_1 和 s_2 在合式公式 φ 中自由出现的所有变量上的取值相同，那么 $\mathfrak{A}$ 以 s_1 满足 φ 当且仅当以 s_2 满足 φ.

情形 1：$\varphi = Pt_1 \cdots t_n$ 是原子公式，那么所有变量在 φ 中自由出现. 这样 s_1 和 s_2 在 φ 中每个变量 t_i 上的取值相同. 于是，对每个 t_i，$s_1(t_i) = \bar{s}_2(t_i)$；详细证明使用对 t_i 的归纳法. 这样就得到，$\mathfrak{A}$ 以 s_1 满足 φ 当且仅当以 s_2 满足 φ.

情形 2 和情形 3：φ 具有 $\neg\,\alpha$ 或者 $\alpha \to \beta$ 的形式. 可以由归纳假设直接得到.

情形 4: $\varphi = \forall x\psi$. 除 x 外, 所有在 φ 中自由出现的变量都是在 ψ 中自由出现的. 对 $|\mathfrak{A}|$ 中任意的 d, $s_1(x|d)$ 和 $s_2(x|d)$ 在 ψ 中自由出现的变量上取值相同. 使用归纳假设, $\mathfrak{A}$ 以 $s_1(x|d)$ 满足 ψ 当且仅当以 $s_2(x|d)$ 满足 ψ. 由此可以看出, $\mathfrak{A}$ 以 s_1 满足 $\forall x\psi$ 当且仅当以 s_2 满足 $\forall x\psi$. ■

实际上, 上述证明要求充分理解满足的定义和 s 提供的信息中哪些是有用的. 对于结构也有类似的结论: 如果 $\mathfrak{A}$ 与 $\mathfrak{B}$ 对所有在 φ 中出现的参数都相同, 那么 $\models_{\mathfrak{A}} \varphi[s]$ iff $\models_{\mathfrak{B}} \varphi[s]$.

该定理说明下述记法是合理的: 假定 φ 是一个公式, 所有在 φ 中自由出现的变量包含在 $v_1, \cdots, v_k$ 中, 那么对 $|\mathfrak{A}|$ 中的元素 $a_1, \cdots, a_k$,

$$\models_{\mathfrak{A}} \varphi[\![a_1, \cdots, a_k]\!]$$

86

意味着 $\mathfrak{A}$ 以某个函数 $s: V \to |\mathfrak{A}|$ 满足 φ, 其中 $s: V \to |\mathfrak{A}|$, 对于 $1 \leqslant i \leqslant k$, 有 $s(v_i) = a_i$. 在上一个例子中, $\mathfrak{A} = (\mathbb{N}; \leqslant, S, 0)$, 有 $\not\models_{\mathfrak{A}} \forall v_2 P v_1 v_2[\![0]\!]$, 但 $\not\models_{\mathfrak{A}} \forall v_2 P v_1 v_2[\![5]\!]$.

推论 22B 对每个句子 σ, 以下两条二者必居其一:

(a) $\mathfrak{A}$ 以每个从 V 到 $|\mathfrak{A}|$ 中的函数 S 满足 σ.

(b) $\mathfrak{A}$ 无法以任何一个从 V 到 $|\mathfrak{A}|$ 中的函数满足 σ.

如果 (a) 成立, 那么我们称 σ 在 $\mathfrak{A}$ 中是*真的*(记作 $\models_{\mathfrak{A}} \sigma$) 或者 $\mathfrak{A}$ 是 σ 的*模型*. 如果 (b) 成立, 那么 σ 在 $\mathfrak{A}$ 中是*假的*. (由于 $\mathfrak{A}$ 是非空的, 两条不能够同时成立.) $\mathfrak{A}$ 是句子集合 Σ 的*模型*当且仅当 $\mathfrak{A}$ 是 Σ 中每个句子的模型.

例 如果 $\mathfrak{R}$ 是实数域 $(\mathbb{R}; 0, 1, +, \times)$, $\mathfrak{Q}$ 是有理数域 $(\mathbb{Q}; 0, 1, +, \times)$, 那么是否存在一个句子在一个域中是真的, 而在另一个域中是假的呢? 答案是肯定的. 因为 $\sqrt{2}$ 是无理数, 句子 $\exists x(x \cdot x = 1 + 1)$ 在有理数域中是假的, 但在实数域中则是真的.

例 假定我们的语言中仅有参数 $\forall$ 和 P, 其中 P 是二元谓词符号, 那么结构 $\mathfrak{A}$ 由 $|\mathfrak{A}|$ 和二元关系 $P^{\mathfrak{A}}$ 确定. 不规范地, 我们可以记作:

$$\mathfrak{A} = (|\mathfrak{A}|; P^{\mathfrak{A}}).$$

考虑如何刻画如下句子的所有模型类:

(1) $\forall x \forall y x = y$. 结构 $(A; R)$ 是该合式公式的一个模型当且仅当 A 仅包含一个元素, R 或者是空集或者是 $A \times A$.

(2) $\forall x \forall y Pxy$. 结构 $(A; R)$ 是该合式公式的一个模型当且仅当 $R = A \times A$, A 可以是任何非空集合.

(3) $\forall x \forall y \neg Pxy$. 结构 $(A; R)$ 是该合式公式的一个模型当且仅当 $R = \varnothing$.

(4) $\forall x \exists y Pxy$. 结构 $(A; R)$ 是该合式公式的一个模型的条件为 R 的定义域是 A.

为方便起见, 采用如下合理的记法:

(1) $\models_{\mathfrak{A}} (\alpha \wedge \beta)[s]$ 当且仅当 $\models_{\mathfrak{A}} \alpha[s], \models_{\mathfrak{A}} \beta[s]$, 对 $\forall$ 和 $\leftrightarrow$ 有类似的记法.

(2) $\models_{\mathfrak{A}} \exists x\alpha[s]$ 当且仅当存在某个 $d \in |\mathfrak{A}|$, 且具有属性 $\models_{\mathfrak{A}} \alpha[s(x|d)]$.

87

其中第 2 个的证明如下:

$$
\begin{aligned}
\vDash_{\mathfrak{A}} \exists x\alpha[s] \quad & \text{iff} \quad \vDash_{\mathfrak{A}} \neg \forall x \neg \alpha[s], \\
& \text{iff} \quad \nvDash_{\mathfrak{A}} \forall x \neg \alpha[s], \\
& \text{iff} \quad \text{对 } |\mathfrak{A}| \text{ 中所有的 } d \text{ 下式不成立:} \\
& \qquad\quad \vDash_{\mathfrak{A}} \neg \alpha[s(x|d)], \\
& \text{iff} \quad \text{对 } |\mathfrak{A}| \text{ 中所有的 } d \text{ 下式不成立:} \\
& \qquad\quad \nvDash_{\mathfrak{A}} \alpha[s(x|d)], \\
& \text{iff} \quad \text{对 } |\mathfrak{A}| \text{ 中的某个 } d,\ \vDash_{\mathfrak{A}} \alpha[s(x|d)] \text{ 成立.}
\end{aligned}
$$

2.2.1 逻辑蕴涵

目前我们已经具备了必要的知识，下面对形式语言中的逻辑蕴涵的重要概念进行公式化描述.

定义 设 Γ 是合式公式的集合，φ 是一个合式公式，那么 Γ*逻辑蕴涵*φ，记作 $\Gamma \vDash \varphi$，当且仅当对语言的每个结构 $\mathfrak{A}$ 和每个函数 $s: V \to |\mathfrak{A}|$，使得 $\mathfrak{A}$ 以 s 满足 Γ 的每个元素，$\mathfrak{A}$ 也以 s 满足 φ.

这里我们使用了和第 1 章中表示重言蕴涵同样的符号"$\vDash$". 但是，此符号今后仅用于表示逻辑蕴涵. 如前，我们用"$\gamma \vDash \varphi$"代替"$\{\gamma\} \vDash \varphi$". 称 φ 与 ψ*逻辑等价* ($\varphi \vDash\!\dashv \psi$) 当且仅当 $\varphi \vDash \psi$ 且 $\psi \vDash \varphi$.

重言式的概念在一阶语言中就是恒真公式的概念. 一个合式公式 φ 是*恒真的* 当且仅当 $\varnothing \vDash \varphi$ (简写作: $\vDash \varphi$). 这样，φ 是恒真的当且仅当对每个结构 $\mathfrak{A}$ 和每个函数 $s: V \to |\mathfrak{A}|$，$\mathfrak{A}$ 以 s 满足 φ.

对于句子，逻辑蕴涵的概念通过使用定理 22A 可以清楚地表达.

推论 22C 对句子集合 $\Sigma; \tau, \Sigma \vDash \tau$ 当且仅当 Σ 的每个模型也是 τ 中的模型. 句子 τ 是恒真的当且仅当其在每个结构下都是真的.

例 (逻辑蕴涵) 读者可以自行判定如下每一条的正误:

(1) $\forall v_1 Q v_1 \vDash Q v_2$

(2) $Q v_1 \nvDash \forall v_1 Q v_1$. 这里只需要找到一个满足如下条件的结构 $\mathfrak{A}$ 和一个函数 $s: V \to |\mathfrak{A}|$ 就够了，一方面 $\vDash_{\mathfrak{A}} Q v_1[s]$，另一方面，$\mathfrak{A}$ 不是 $\forall v_1 Q v_1$ 的模型. $|\mathfrak{A}|$ 中至少需要两个元素.

88

(3) $\vDash \neg\neg\sigma \to \sigma$. 如果 $\mathfrak{A}$ 是 $\neg\neg\sigma$ 的模型，那么 $\nvDash_{\mathfrak{A}} \neg\sigma, \vDash_{\mathfrak{A}} \sigma$. 可能会有人问这样的问题：我们是不是使用了正要证明的双重否定律？答案是明确的. 我们要证明的是形式语言 (有时称为 *对象语言*) 的双重否定律. 为此，我们当然可以使用任何正确的推理 (元语言，自然语言)，正如在向量空间或者图论中的推理一样. 特别地，在形式模型中，推理可能会涉及包含 $\neg\neg\sigma$ 与 σ 的原则. 这不是循环论证. 然而，毫不奇怪，我们使用的元语言中的命题是与对象语言的公式相关的. 关于这个关系，可以参见 2.4 节结尾的图.

(4) $\forall v_1 Q v_1 \vDash \exists v_2 Q v_2$. 任何结构的域都是非空的.

(5) $\exists x \forall y Pxy \vDash \forall y \exists x Pxy$. 这个例子在 2.4 节还会出现.

(6) $\forall y \exists x Pxy \not\vDash \exists x \forall y Pxy$.

(7) $\vDash \exists x(Qx \to \forall x Qx)$. 这个句子有些奇怪，但是是恒真的.

逻辑蕴涵的定义与第 1 章中的重言蕴涵非常相似，但是其复杂性却大大不同. 在命题逻辑中想要判定一个合式公式是否是重言式，只需要考虑有限个真值指派，其中每一个都是有限函数. 对每个真值指派，只要计算 $\bar{v}(\alpha)$，这可以在有限长的时间内完成. (如前所述，重言式的集合是可以判定的.)

与之相反，如果要判定一阶语言中的合式公式是否恒真，则需要考虑每一个结构 $\mathfrak{A}$. (特别地，这需要使用每个非空集合，而每个集合都有很多元素.) 对于每个结构，还要考虑每个从变量集合 V 到 $|\mathfrak{A}|$ 的函数 s. 并且对每一个给定的结构 $\mathfrak{A}$ 和 s，还需要判定 $\mathfrak{A}$ 是否以 s 满足 φ. 如果 $|\mathfrak{A}|$ 是无限的，其本身就是一个非常复杂的概念.

考虑到这些复杂性，恒真公式集的无法判定性就不足为怪了 (见 3.5 节). 奇怪的是，可以证明恒真性的概念与另外一个概念 (可推导性) 是等价的，这个概念更接近有限性 (见 2.4 节). 利用这个等价，(在某些推理假设下) 可以证明恒真集 (恒真合式公式的集合) 事实上是可数的. 由这个可数的性质可以得到恒真公式集的一个更具体的特性.

89

2.2.2 结构中的可定义性

现在来看实数域 ($\mathbb{R}$; 0, 1, +, ·)，其中包含实数集、两个不同元素 0 与 1 和两种运算：加法与乘法. 将实数域看作一个结构：

$$\mathfrak{R} = (\mathbb{R}; 0, 1, +, \cdot),$$

而语言含有常数符号 **0** 与 **1** 以及二元函数符号 + 与 ·.

尽管其不含有序符号 <，事实上可以使用另外一种方式表达 "$x \geqslant 0$" . 这是由于 *在此结构中，非负数有平方根*. 即只要 x 的取值为非负数，结构就可以满足公式 $\exists v_2 x = v_2 \cdot v_2$，这就是说，

$$\vDash_{\mathfrak{R}} \exists v_2 v_1 = v_2 \cdot v_2 [\![a]\!] \Leftrightarrow a \geqslant 0.$$

由此，可以说区间 $[0, \infty)$ 在结构 $\mathfrak{R}$ 中是 *可定义的*，用公式 $\exists v_2 v_1 = v_2 \cdot v_2$ 就可以定义.

另外，实数中的序关系，即二元关系

$$\{\langle a, b\rangle \in \mathbb{R} \times \mathbb{R} | a \leqslant b\},$$

可以由结构 $\mathfrak{R}$ 中的用于表达 "$v_1 \leqslant v_2$" 的公式定义：

$$\exists v_3 v_2 = v_1 + v_3 \cdot v_3.$$

作为一个简单的例子，来看一下有向图

$$\mathfrak{A} = (\{a, b, c\}; \{\langle a, b\rangle, \langle a, c\rangle\})$$

其中语言含有参数 $\forall$ 和 E：

$$b \bullet \longleftarrow \overset{a}{\bullet} \longrightarrow \bullet\, c$$

那么在 $\mathfrak{A}$ 中，集合 $\{b,c\}$(关系 $E^{\mathfrak{A}}$ 的范围) 定义为公式 $\exists v_2 E v_2 v_1$. 反之，集合 $\{b\}$ 在 $\mathfrak{A}$ 中是不 能够定义的，这是因为在此结构中没有将 b 与 c 分开的可定义属性，这一点在本节稍后使用同态定理进行验证.

现在我们要简化论域 (或论域上关系) 的子集的可定义性的概念. 考虑结构 $\mathfrak{A}$ 与合式公式 φ，其自由变元在 $v_1, \cdots, v_k$ 之中，那么可以构建 $|\mathfrak{A}|$ 上的 k 元关系

$$\{\langle a_1, \cdots, a_k\rangle | \models_{\mathfrak{A}} \varphi[\![a_1, \cdots, a_k]\!]\}.$$

90

称此 k 元关系是在 $\mathfrak{A}$ 中由 φ*定义* 的. 一般地，$|\mathfrak{A}|$ 上的 k 元关系在 $\mathfrak{A}$ 中是 *可定义的* 当且仅当存在能够定义它的一个公式 (其自由变元在 $v_1, \cdots, v_k$ 中).

例　考虑数论语言的一部分，指定语言中有参数 $\forall$, $\mathbf{0}$, $\mathbf{S}$, $+$ 和 $\cdot$. 令 $\mathfrak{N}$ 是如下结构：

$|\mathfrak{N}| = \mathbb{N}$，自然数集.

$\mathbf{0}^{\mathfrak{N}} = 0$，数字 0.

$\mathbf{S}^{\mathfrak{N}}$，$+^{\mathfrak{N}}$，$\cdot^{\mathfrak{N}}$ 是 S，$+$，$\cdot$，后继函数，加法和乘法.

用等式表示为

$$\mathfrak{N} = (\mathbb{N}; 0, S, +, \cdot).$$

$\mathbb{N}$ 上的有些关系在 $\mathfrak{N}$ 中是可定义的，有些是不可定义的. 证明不可定义性的一个方法要用到在 $\mathbb{N}$ 上有不可数多个关系，而仅有可数个可以定义的公式. (然而，给一个特定的例子是非常困难的. 毕竟，如果某个东西是不可定义的，也就确实难以确切地说明它是什么! 在 3.5 节，我们会看到一个特定的例子：在 $\mathfrak{N}$ 中为真的句子的哥德尔数的集合.)

(1) 序关系 $\{\langle m,n\rangle | m < n\}$ 在 $\mathfrak{N}$ 中由如下公式定义：

$$\exists v_3 v_1 + \mathbf{S} v_3 = v_2.$$

(2) 对任意的自然数 n，$\{n\}$ 是可定义的，如 $\{2\}$ 可定义为

$$v_1 = \mathbf{SS0}.$$

因此，我们称 n 是 $\mathfrak{N}$ 中 *可定义的元素*.

(3) 素数集在 $\mathfrak{N}$ 中是可定义的. 如果对符号 1 和 $<$ 有参数 $\mathbf{1}$ 和 $<$，素数集可由如下公式定义：

$$\mathbf{1} < v_1 \wedge \forall v_2 \forall v_3 (v_1 = v_2 \cdot v_3 \rightarrow v_2 = \mathbf{1} \vee v_3 = \mathbf{1}).$$

但由于 $\{1\}$ 和 $<$ 在 $\mathfrak{N}$ 中是可定义的，无需为它们添加参数，可以直接使用其定义. 这样素数集可定义为

$$\exists v_3 \mathbf{S0} + \mathbf{S} v_3 = v_1 \wedge \forall v_2 \forall v_3 (v_1 = v_2 \cdot v_3 \rightarrow v_2 = \mathbf{S0} \vee v_3 = \mathbf{S0}).$$

(4) 幂运算，$\{\langle m,n,p\rangle | p = m^n\}$ 在 $\mathfrak{N}$ 中也是可定义的. 这一点似乎不太容易看出来，在

91

3.8 节，我们将会使用中国剩余定理给出证明.

实际上，后面我们会证明任意的 $\mathbb{N}$ 上可以判定的关系在 $\mathfrak{N}$ 中都是可定义的，任何能行可枚举关系和许多其他关系也是可定义的. 在某种程度上，可定义关系的复杂度可由最简单定义的关系进行度量. 在 3.5 节的结尾处，这个思路会再次用到.

2.2.3 结构类的可定义性

在开始讲述数学中的类的时候，老师往往会讲类似下面的说法：

(1) “*图* 定义为包括非空集合 V 和集合 E，满足……”

(2) “*群* 定义为包括非空集合 G 和二元关系 $\circ$，满足如下公理……”

(3) “*有序域* 定义为包括非空集合 F 和两个二元运算 $+$ 与 $\cdot$，以及二元关系 $<$，满足如下公理……”

(4) “*向量空间* 定义为包括非空集合 V 和二元运算 $+$，以及对每个实数 r 的数量乘法，满足……”

我们希望对这些情况进行抽象．在每种情况中，所研究的对象 (图、群等) 都是某种合适的语言的 *结构*．而且，这些对象都需要满足某个句子集合 Σ(即“公理”)．这些内容涉及的课程主要研究公理集的模型——至少是某些模型．

对句子集 Σ，用 $\text{Mod}\ \Sigma$ 表示 Σ 的所有模型组成的类，即某种语言的所有结构的类，在这种语言中 Σ 的每个元素都是真的．对单个句子 τ，我们简单地记作 $\text{Mod}\ \tau$，而不用 $\text{Mod}\ \{\tau\}$．(熟悉公理集合论的读者应该注意到，$\text{Mod}\ \Sigma$ 如果是非空的，那么它就是正则类；它可能太大而不能成为一个集合．)

我们的语言的结构类 $\mathcal{K}$ 是 *初等类*(elementary class, EC) 当且仅当对某个句子 τ，$\mathcal{K} = \text{Mod}\ \tau$．$\mathcal{K}$ 是 *广义初等类*(elementary class in wider sense, EC_Δ) 当且仅当对某个句子集 Σ，$\mathcal{K} = \text{Mod}\ \Sigma$．(其中“初等”与“一阶”是同义词．)

例 (1) 假定语言有相等以及两个参数 $\forall$ 和 E，其中 E 是一个二元谓词符号．那么 *图* 是该语言的一个结构 $\mathfrak{A} = (V; E^{\mathfrak{A}})$，包括 *顶点*(或者 *节点*) 的非空集合 V 和 *边关系* $E^{\mathfrak{A}}$，如果满足只要 $uE^{\mathfrak{A}}v$ 就有 $vE^{\mathfrak{A}}u$，该关系就是对称的，如果不满足 $vE^{\mathfrak{A}}v$，则该关系是非自反的．用于说明边关系是对称的和非自反的公理可以由如下句子表达：

92

$$\forall x(\neg\, xEx \wedge \forall y(xEy \to yEx)).$$

因此所有图的类是初等类．对 *有向图*，对称的前提可以去掉．如果允许出现“环”那么非自反的前提也可以去掉．但是，老师或许会说这个课程只讨论 *有限图*．所有有限图的类是初等类吗? 不是的，后面我们会证明这一点，即使在广义的情况下也不是．

(2) 假定语言有相等以及两个参数 $\forall$ 和 P，其中 P 是一个二元谓词符号．如前述，该语言的结构 $(A; R)$ 包括非空集合 A 和 A 上的二元关系 R．$(A; R)$ 称为 *有序集* 当且仅当 R 是传递的，且满足 *三分律* (trichotomy)．(三分律是指对于 A 中的元素 a 与 b，$\langle a, b\rangle \in R$, $a = b$ 和 $\langle b, a\rangle \in R$ 三者必有一个成立．) 因为这些条件在形式语言可以翻译为一个句子，所以非空有序集的类是初等类．事实上就是 $\text{Mod}\ \tau$，其中 τ 是如下 3 个句子的合取：

$$\forall x\forall y\forall z(xPy \to yPz \to xPz);$$

$$\forall x\forall y(xPy \vee x = y \vee yPx);$$

$$\forall x\forall y(xPy \to \neg\, yPx).$$

下面的例子假定读者熟悉代数的相关知识．

(3) 假定语言有 $=$ 以及两个参数 $\forall$ 和 $\circ$，其中 $\circ$ 是一个二元谓词符号．所有群的类是初等类，是所有满足如下群的公理的模型的类：

$$\forall x \forall y \forall z (x \circ y) \circ z = x \circ (y \circ z);$$

$$\forall x \forall y \exists z x \circ z = y;$$

$$\forall x \forall y \exists z z \circ x = y.$$

所有无限群的类是 EC_Δ．为了说明这一点，设

$$\lambda_2 = \exists x \exists y x \neq y,$$

$$\lambda_3 = \exists x \exists y \exists z (x \neq y \wedge x \neq z \wedge y \neq z),$$

$$\cdots.$$

这样，λ_n 即是“至少要有 n 个对象”．那么，有了 $\{\lambda_2, \lambda_3, \cdots\}$，群公理形成集合 Σ，其中 $\mathrm{Mod}\,\Sigma$ 恰好是无限群的类．在 2.6 节，我们会证明无限群的类不是 EC．

93

(4) 假定语言有相等以及参数 $\forall$，$\mathbf{0}$，$\mathbf{1}$，$+$，$\cdot$．域可以看作该语言的结构，所有域的类是初等类．特征 0 的域的类是 EC_Δ，而不是 EC，这可以由 2.6 节的一阶语言的紧致性定理推出．

2.2.4　同态[1]

在图论、群或者向量空间的课程中，经常会遇到两个结构同构的概念，$\mathfrak{A}$ 与 $\mathfrak{B}$ 是同构的：大致是说，存在一个从 $|\mathfrak{A}|$ 到 $|\mathfrak{B}|$ 的一一对应，并且这个对应能够保持相应的运算和关系．

两个同构的结构尽管有所不同，但是却具有相同的数学性质．我们希望能够给出同构的一般意义上的概念，以说明两个同构的结构必须满足相同的命题．

设 $\mathfrak{A}$、$\mathfrak{B}$ 是语言的结构．从 $\mathfrak{A}$ 到 $\mathfrak{B}$ 中的一个 同态(homomorphism) 是一个函数 $h : |\mathfrak{A}| \to |\mathfrak{B}|$，具有下列性质：

(a) 对每个 n 元谓词参数 P 和 $|\mathfrak{A}|$ 中元素的任意 n 元组 $\langle a_1, \cdots, a_n \rangle$，

$$\langle a_1, \cdots, a_n \rangle \in P^{\mathfrak{A}} \quad \text{iff} \quad \langle h(a_1), \cdots, h(a_n) \rangle \in P^{\mathfrak{B}}.$$

(b) 对每个 n 元函数符号 f 和任意 n 元组，

$$h(f^{\mathfrak{A}}(a_1, \cdots, a_n)) = f^{\mathfrak{B}}(h(a_1), \cdots, h(a_n)).$$

对于常数符号 c，

$$h(c^{\mathfrak{A}}) = c^{\mathfrak{B}}.$$

条件 (a) 与 (b) 用于说明“h 保持关系和函数”．(需要指出的是，有些书中使用了条件 (a) 的一个较弱的说法；这里的同态是他们所谓的“强同态”．)

另外，如果 h 是一对一的，那么同态就变成了从 $\mathfrak{A}$ 到 $\mathfrak{B}$ 中的 同构(或者 同构嵌入)．若存在 $\mathfrak{A}$ 到 $\mathfrak{B}$ 上的同构 (即同构 h 满足 $h = |\mathfrak{B}|$)，那么 $\mathfrak{A}$ 和 $\mathfrak{B}$ 称作是 同构的(记作 $\mathfrak{A} \cong \mathfrak{B}$)．

1. 这部分内容可以推后学习．但在 (含有等号的) 完全性定理的证明中需要用到同态，在第 2 章的 2.6 节中我们会用到同构的概念．

读者以前可能会遇到这个概念在群与域中的一些特例.

94

例 假定语言中的参数有 $\forall$, + 和 ·, 令 $\mathfrak{A}$ 是结构 $(\mathbb{N};+,\cdot)$, 定义函数 $h:\mathbb{N}\to\{e,o\}$ 如下:

$$h(n)=\begin{cases} e & \text{如果 } n \text{ 为偶数,}\\ o & \text{如果 } n \text{ 为奇数.}\end{cases}$$

那么 h 是从 $\mathfrak{A}$ 到 $\mathfrak{B}$ 上的同态, 其中 $|\mathfrak{B}|=\{e,o\}$ 且 $+^{\mathfrak{B}}$ 和 $\cdot^{\mathfrak{B}}$ 定义如下:

$+^{\mathfrak{B}}$	e	o
e	e	o
o	o	e

$\cdot^{\mathfrak{B}}$	e	o
e	e	e
o	e	o

可以证明定义中的条件 (b) 是可以满足的. 例如, 若 a 与 b 都是奇数, 那么 $h(a+b)=e$, 而 $h(a)+^{\mathfrak{B}}h(b)=o+^{\mathfrak{B}}o=e$.

例 设 $\mathbb{P}$ 是正整数的集合, $<_P$ 是 $\mathbb{P}$ 上通常的有序关系, $<_N$ 是 $\mathbb{N}$ 上的通常的有序关系. 那么存在从结构 $(\mathbb{P};<_P)$ 到 $(\mathbb{N};<_N)$ 上的同构 h; 可取 $h(n)=n-1$. 同样地, 恒等映射 $Id:\mathbb{P}\to\mathbb{N}$ 也是一个从结构 $(\mathbb{P};<_P)$ 到 $(\mathbb{N};<_N)$ 上的同构. 因此, 称 $(\mathbb{P};<_P)$ 是 $(\mathbb{N};<_N)$ 的*子结构*.

更一般地, 考虑两个结构 $\mathfrak{A}$ 与 $\mathfrak{B}$, 这里 $|\mathfrak{A}|\subseteq|\mathfrak{B}|$. 显然, 由同态的定义可以看出, 从 $|\mathfrak{A}|$ 到 $|\mathfrak{B}|$ 中的恒等映射是从 $\mathfrak{A}$ 到 $\mathfrak{B}$ 中的同态当且仅当

(a) 对每个谓词 P, $P^{\mathfrak{A}}$ 是 $P^{\mathfrak{B}}$ 在 $|\mathfrak{A}|$ 上的限制;

(b) 对每个函数 f, $f^{\mathfrak{A}}$ 是 $f^{\mathfrak{B}}$ 在 $|\mathfrak{A}|$ 上的限制, 且对每个常数符号 c, 有 $c^{\mathfrak{A}}=c^{\mathfrak{B}}$.

如果这些条件都满足了, 那么 $\mathfrak{A}$ 就是 $\mathfrak{B}$ 的*子结构*, $\mathfrak{B}$ 是 $\mathfrak{A}$ 的*扩充*.

例如, 在具有二元函数符号 + 的语言中, 结构 $(\mathbb{Q};+_Q)$ 是 $(\mathbb{C};+_C)$ 的子结构. 这里的 $+_C$ 是复数的加运算; $+_Q$ 是复数上的加运算 $+_C$ 在有理数集 $\mathbb{Q}$ 上的限制.

在此例中, 集合 $\mathbb{Q}$ 在运算 $+_C$ 下是封闭的; 即任意两个有理数的和还是有理数. 更一般地, 只要 $\mathfrak{A}$ 是 $\mathfrak{B}$ 的子结构, 那么对每个函数符号 f, $|\mathfrak{A}|$ 在运算 $f^{\mathfrak{B}}$ 下必定是封闭的. 毕竟, $f^{\mathfrak{B}}(\vec{a})$ 就是 $|\mathfrak{A}|$ 中的某个元素 (其中 $\vec{a}\in|\mathfrak{A}|^n$). 这一封闭性对 0 元函数符号也是成立的; 对每个常数符号 c, $c^{\mathfrak{B}}$ 肯定属于 $|\mathfrak{A}|$.

95

反之, 设有结构 $\mathfrak{B}$, 令 A 是 $|\mathfrak{B}|$ 的任意非空子集, $\mathfrak{A}$ 在所有 $\mathfrak{B}$ 的函数作用下都是封闭的. 那么取 $\mathfrak{B}$ 的子结构, 其论域为 $\mathfrak{A}$. 事实上, 这个结构是唯一的. 论域为 $\mathfrak{A}$, 其每个谓词参数 P 被指定为 $P^{\mathfrak{B}}$ 在 $\mathfrak{A}$ 上的限制, 函数符号也是如此. 作为一个极端的例子, 如果语言中没有函数符号 (也没有常数符号), 那么可以对 $|\mathfrak{B}|$ 中的*任意*非空子集 A 做一个子结构.

同态定理 设 h 是从 $\mathfrak{A}$ 到 $\mathfrak{B}$ 中的同态, s 将变量的集合映射到 $|\mathfrak{A}|$ 中.

(a) 对每个项 t, 我们有 $h(\bar{s}(t))=\overline{h\circ s}(t)$, 其中 $\bar{s}(t)$ 是在 $\mathfrak{A}$ 中计算的, 而 $\overline{h\circ s}(t)$ 是在 $\mathfrak{B}$ 中计算的;

(b) 对每个不包含等于符号的无量词的公式 α,

$$\models_{\mathfrak{A}}\alpha[s] \text{ iff } \models_{\mathfrak{B}}\alpha[h\circ s].$$

(c) 如果 h 是一个一对一的 (即是从 $\mathfrak{A}$ 到 $\mathfrak{B}$ 中的同构)，那么在 (b) 中我们可以去掉“不包含等于符号”的限制；

(d) 如果 h 是 $\mathfrak{A}$ 到 $\mathfrak{B}$ 上的同态，那么在 (b) 中可以去掉“无量词”的限制.

证明　(a) 使用对 t 的归纳法；见习题 13. 注意 $h \circ s$ 将变量集映射到 $|\mathfrak{B}|$ 中；将其扩充到所有项的集合就是 $\overline{h \circ s}$，而它正是在 t 的取值.

(b) 对原子公式，如 Pt，有

$$
\begin{aligned}
\models_{\mathfrak{A}} Pt[s] &\Leftrightarrow \bar{s}(t) \in P^{\mathfrak{A}} \\
&\Leftrightarrow h(\bar{s}(t)) \in P^{\mathfrak{B}} \quad \text{因为 } h \text{ 是同态} \\
&\Leftrightarrow \overline{h \circ s}(t) \in P^{\mathfrak{B}} \quad \text{用}(a) \\
&\Leftrightarrow \models_{\mathfrak{B}} Pt[h \circ s].
\end{aligned}
$$

使用归纳证明可以处理联结符号 $\neg$ 与 $\rightarrow$，完全按照程序做就行了.

(c) 在任意情况下，

$$
\begin{aligned}
\models_{\mathfrak{A}} u = t[s] &\Leftrightarrow \bar{s}(u) = \bar{s}(t) \\
&\Rightarrow h(\bar{s}(u)) = h(\bar{s}(t)) \\
&\Leftrightarrow \overline{h \circ s}(u) = \overline{h \circ s}(t) \quad \text{用}(a) \\
&\Leftrightarrow \models_{\mathfrak{B}} u = t[h \circ s].
\end{aligned}
$$

如果 h 是一对一的，第 2 步的箭头可以反过来.

(d) 将 (b) 中的归纳法扩充到包括量词的情况，即我们需要证明，如果对每个 s，φ 都具

96

有如下性质：

$$\models_{\mathfrak{A}} \varphi[s] \Leftrightarrow \models_{\mathfrak{B}} \varphi[h \circ s],$$

那么 $\forall x \varphi$ 满足同样的性质. (作为对 φ 的推理假设的结论) 在任何情况下，都有蕴涵式

$$\models_{\mathfrak{B}} \forall x \varphi[h \circ s] \Rightarrow \models_{\mathfrak{A}} \forall x \varphi[s].$$

直观上看，这是很可靠的；在一个大的集合 $|\mathfrak{B}|$ 中，如果 φ 是真的，那么在一个较小的集合中肯定也是真的. 详细地说，对 $|\mathfrak{A}|$ 中的每个元素 a，

$$
\begin{aligned}
\models_{\mathfrak{B}} \forall x \varphi[h \circ s] &\Rightarrow \models_{\mathfrak{B}} \varphi[(h \circ s)(x|h(a))] \quad \text{由归纳假设，函数的情况也是相同的} \\
&\Leftrightarrow \models_{\mathfrak{B}} \varphi[(h \circ (s(x|a))], \\
&\Leftrightarrow \models_{\mathfrak{A}} \varphi[s(x|a)]
\end{aligned}
$$

现在为方便起见，假定 $\not\models_{\mathfrak{B}} \forall x \varphi[h \circ s]$，那么对 $|\mathfrak{B}|$ 中每个元素 b，$\models_{\mathfrak{B}} \neg \varphi[(h \circ s)(x|b)]$. 我们需要如下蕴涵式：

(*) 如果对某个 $|\mathfrak{B}|$ 中的 b，$\models_{\mathfrak{B}} \neg\varphi[(h \circ s)(x|b)]$，

那么对对某个 $|\mathfrak{A}|$ 中的 a，$\models_{\mathfrak{B}} \neg \varphi[(h \circ s)(x|h(a))]$.

对于给定 (*)，有

$$\begin{aligned}\models_{\mathfrak{B}} \neg\varphi[(h\circ s)(x|h(a))] &\Leftrightarrow \models_{\mathfrak{B}} \neg\varphi[h\circ(s(x|a))], \quad \text{由归纳假设，函数的情况也是相同的}\\ &\Leftrightarrow \models_{\mathfrak{A}} \neg\varphi[s(x|a)]\\ &\Rightarrow \not\models_{\mathfrak{A}} \forall x\varphi[s].\end{aligned}$$

如果 h 将 $|\mathfrak{A}|$ 映射到 $|\mathfrak{B}|$ 上，那么 (*) 是显而易见的；取 a 满足 $b=h(a)$. (即使 h 的值不在 $|\mathfrak{B}|$ 中，(*) 也可能会成立.) ■

称语言的两个结构 $\mathfrak{A}$ 与 $\mathfrak{B}$ 是*初等等价的*(elementarily equivalent)(记作 $\mathfrak{A}\equiv\mathfrak{B}$) 当且仅当对任意的句子 σ,

$$\models_{\mathfrak{A}} \sigma \Leftrightarrow \models_{\mathfrak{B}} \sigma.$$

推论 22D 同构的结构是初等等价的：

$$\mathfrak{A}\cong\mathfrak{B}\Rightarrow\mathfrak{A}\equiv\mathfrak{B}$$

实际上不止如此，同构的结构在“结构”的每个方面都是相似的, 它们不只满足相同的一阶命题，也满足相同的二阶 (和更高阶的) 命题 (即它们是二阶等价的，甚至更高阶等价).

有些初等等价的结构是不同构的. 例如，可以证明包含实数集 (具有通常的序关系) 的结构 $(\mathbb{R};<_R)$ 初等等价于包含有理数集 (具有序关系) 的结构 $(\mathbb{Q};<_Q)$(见 2.6 节). 但是 $\mathbb{Q}$ 是可数集，而 $\mathbb{R}$ 是不可数的，因此，二者肯定是不同构的. 在 2.6 节，我们会很容易地明白不同基数的结构为什么初等等价. 97

例 再看前面的例子. 设有从 $(\mathbb{P};<_P)$ 到 $(\mathbb{N};<_N)$ 上的同构 h. 特别地，$(\mathbb{P};<_P)\equiv(\mathbb{N};<_N)$；这些结构在一阶语言中无法通过句子进行区分.

进一步，注意到恒等映射是从 $(\mathbb{P};<_P)$ 到 $(\mathbb{N};<_N)$ 中的同构嵌入. 因此，对函数 $s:V\to\mathbb{P}$ 和无量词的 φ,

$$\models_{(\mathbb{P};<_P)} \varphi[s] \Leftrightarrow \models_{(\mathbb{N};<_N)} \varphi[s].$$

当 φ 包含量词时，这个等价就不成立了. 例如，

$$\models_{(\mathbb{P};<_P)} \forall v_2(v_1\neq v_2\to v_1<v_2)[\![1]\!],$$

但是

$$\not\models_{(\mathbb{N};<_N)} \forall v_2(v_1\neq v_2\to v_1<v_2)[\![1]\!].$$

结构 $\mathfrak{A}$ 的一个*自同构* (automorphism) 是从 $\mathfrak{A}$ 到 $\mathfrak{A}$ 上的一个同构. $\mathfrak{A}$ 上的恒等函数很显然是一个 $\mathfrak{A}$ 的自同构. $\mathfrak{A}$ 可能会有非平凡的自同构，也可能没有. (如果恒等函数是唯一的自同构，则称 $\mathfrak{A}$ 是*固化的*.) 由同态定理，可以证明自同构一定保持可定义的关系：

推论 22E 设 h 是 $\mathfrak{A}$ 的自同构，R 是 $|\mathfrak{A}|$ 上的 n 元关系，那么对 $|\mathfrak{A}|$ 中任意的 $a_1,\cdots,a_n$,

$$\langle a_1,\cdots,a_n\rangle\in R\Leftrightarrow\langle h(a_1),\cdots,h(a_n)\rangle\in R.$$

证明　设 φ 是在 $\mathfrak{A}$ 中定义 R 的一个公式，我们需要知道

$$\models_{\mathfrak{A}} \varphi[\![a_1,\cdots,a_n]\!] \Leftrightarrow \models_{\mathfrak{A}} \varphi[\![h(a_1),\cdots,h(a_n)]\!].$$

由同态定理，这是显然的. ■

这个推论在证明给定的关系不可定义时可能会有用. 例如，考虑结构 $(\mathbb{R};<)$，包含具有通常序关系的实数. 该结构的自同构不过是一个从 $\mathbb{R}$ 到 $\mathbb{R}$ 的函数 h，这个函数是严格递增

98

的：

$$a<b \Leftrightarrow h(a)<h(b).$$

满足该条件的一个函数是 $h(a)=a^3$. 因为这个函数将不属于 $\mathbb{N}$ 的点映射到 $\mathbb{N}$ 中了，集合 $\mathbb{N}$ 在此结构中是不可定义的.

另外一个例子是在初等代数课本给出的，其中讲到平面中的向量的长度不能使用向量加法和数量乘法进行定义. 从向量 $\mathbf{x}$ 到 $\mathbf{2x}$ 中的映射是平面的一个自同构，但是不能够保持其长度. 从这个角度来看，讨论的结构，

$$(E;+,f_r)_{r\in\mathbb{R}},$$

其论域为平面 E，具有二元向量加法运算 $+$，以及 (对 $\mathbb{R}$ 中的每个 r) 由 r 确定的一元数量乘法运算 f_r (这样所涉及的语言就对于每个实数有一个一元函数符号.) 上面所述的双射是这个结构的自同构，但是它不能保持单位向量的集合，

$$\{\mathbf{x}|\mathbf{x}\in E\text{且 }\mathbf{x}\text{ 的长度为 }1\}$$

故在这个结构中，该集合是不可定义的. (附注：向量空间的这个同态称为 *线性变换*.)

习题

1. 证明：(a) $\Gamma;\alpha\models\varphi$　iff　$\Gamma\models(\alpha\to\varphi)$, (b) $\varphi\models\!\!\!=\!\!\!\mid\psi$　iff　$\models(\varphi\leftrightarrow\psi)$.
2. 证明：下面的句子中的任何一个都不能由另外两个逻辑蕴涵. (*提示*：给定一个结构，在此结构中一个句子是假的，而其他两个是真的.)
   (a) $\forall x\forall y\forall z(Pxy\to Pyz\to Pxz)$.方便记法$\alpha\to\beta\to\gamma$ 是$\alpha\to(\beta\to\gamma)$
   (b) $\forall x\forall y(Pxy\to Pyx\to x=y)$
   (c) $\forall x\exists yP_{xy}\to\exists y\forall xPxy$
3. 证明：
$$\{\forall x(\alpha\to\beta),\forall x\alpha\}\models\forall x\beta.$$
4. 证明：如果 x 在 α 中不是自由出现的，那么 $\alpha\models\forall x\alpha$.
5. 证明：公式 $x=y\to Pzfx\to Pzfy$ 是恒真的 (其中 f 是一元函数符号，P 是二元谓词符号).
6. 证明：公式 θ 是恒真的当且仅当 $\forall x\theta$ 是恒真的.

99

7. 用 84 页描述的方法重述"$\mathfrak{A}$ 以 s 满足 φ"的定义，即递归定义函数 $\bar{h}$，使得 $\mathfrak{A}$ 以 s 满足 φ 当且仅当 $s\in\bar{h}(\varphi)$.
8. 设 Σ 是句子集合，且满足对于任意的句子，或者 $\Sigma\models\tau$ 或者 $\Sigma\models\neg\tau$. 假设 $\mathfrak{A}$ 是 Σ 的模型. 证明对任意的句子 τ，有 $\models_{\mathfrak{A}}\tau$ 当且仅当 $\Sigma\models\tau$.
9. 设语言具有相等和二元谓词符号 P. 对下面条件中的每一个，找出一个句子 σ 使得结构 $\mathfrak{A}$ 是 σ 的模型，当且仅当条件被满足.

(a) $|\mathfrak{A}|$ 中恰有两个元素.

(b) $P^{\mathfrak{A}}$ 是从 $|\mathfrak{A}|$ 到 $|\mathfrak{A}|$ 中的函数. (如第 1 章中所述, 函数是单值关系. 对从 $\mathfrak{A}$ 到 $\mathfrak{B}$ 的函数 f, 其定义域为 $\mathfrak{A}$ 的全部, 值域是 $\mathfrak{B}$ 的子集, 但无需是真子集.)

(c) $P^{\mathfrak{A}}$ 是 $|\mathfrak{A}|$ 的一个置换, 即 $P^{\mathfrak{A}}$ 是定义域和值域都是 $|\mathfrak{A}|$ 的一对一函数.

10. 证明:

$$\models_{\mathfrak{A}} \forall v_2 Q v_1 v_2 [\![c^{\mathfrak{A}}]\!] \text{ iff } \models_{\mathfrak{A}} \forall v_2 Q c v_2.$$

其中 Q 是二元谓词符号, c 是常数符号.

11. 给出能够在 $(\mathbb{N};+,\cdot)$ 中定义如下每一个关系的公式. (假设语言具有相等符号和参数 $\forall$, $+$, $\cdot$)

(a) $\{0\}$.

(b) $\{1\}$.

(c) $\{\langle m,n\rangle | n$ 是 $\mathbb{N}$ 中 m 的后继 $\}$.

(d) $\{\langle m,n\rangle |$ 在 $\mathbb{N}$ 中 $m<n\}$.

附注: 该题仅是表面上的描述. 如果一个关系在这个结构中是可定义的, 那么就称为是*算术的*. 所有可以判定的关系都是算术的. 算术关系可以按照层次进行排列; 见 3.5 节.

12. 设 $\mathfrak{R}$ 是结构 $(\mathbb{R};+,\cdot)$.(假设语言具有相等符号和参数 $\forall$, $+$, $\cdot$. $\mathfrak{R}$ 是论域为实数集 $\mathbb{R}$ 的结构, 且 $+^{\mathfrak{R}}, \cdot^{\mathfrak{R}}$ 是通常的加法和乘法运算.)

(a) 给出在 $\mathfrak{R}$ 中定义区间 $[0,\infty)$ 的公式;

(b) 给出在 $\mathfrak{R}$ 中定义集合 $\{2\}$ 的公式;

*(c) 证明端点是代数的任意有限个区间的并在 $\mathfrak{R}$ 中是可定义的. (反之也是成立的; 只有这些集合在 $\mathfrak{R}$ 中是可定义的, 但这一点我们无法证明.)

13. 证明同态定理的 (a).

100

14. 实数集的哪些子集在 $(\mathbb{R};<)$ 中是可定义的? 实平面 $\mathbb{R}\times\mathbb{R}$ 的哪些子集在 $(\mathbb{R};<)$ 中是可定义的?

说明: $(\mathbb{R};<)$ 的一个很好的性质是从 $\mathbb{R}$ 到其自身的序保持映射恰好是自同构的. 但是高于二元的关系就不一定了, 可定义的三元关系有 2^{13} 个, 我们无法将其分类.

15. 证明: 加法关系 $\{\langle m,n,p\rangle | p=m+n\}$ 在 $(\mathbb{N};\cdot)$ 中是不可定义的. 提示: 考虑 $(\mathbb{N};\cdot)$ 的一个自同构, 将两个素数互换.

附注: 从代数性质上看, 带有乘法的自然数不过是具有 $\aleph_0$ 个生成元 (即素数) 和一个非零元的自由阿贝尔半群. 事实上, 我们无法定义加法. 假如能够定义加法, 就能够定义序关系 (由习题 11 和传递性自然也就可定义). 生成元可以看成都是一样的, 即仅仅是素数的置换就产生 $2^{\aleph_0}$ 个自同构. 但是除恒等以外, 没有一个能够保持序关系.

16. 对每一个正整数 n, 给出一个句子, 该句子有大小为 $2n$ 的模型, 但是没有奇数大的模型. (这里的语言应该包括相等和我们可以选择的任意参数.) 提示: 定义句子的方法是 "所有的对象都是红的或者是蓝的, f 是交换颜色的置换. "

说明: 给定句子 σ, 可能会有某些有限模型 (即具有有限论域的模型). 定义 σ 的*谱系*为这样一些正整数 n 的集合: σ 具有大小为 n 的模型. 这个习题证明了偶数集是一个谱系. 例如: 若 σ 是域公理的合取 (只存在有限多个公理, 故可取其合取), 那么其谱系是素数的幂集. 这一点可在任何有限域中证明. 反之, $\neg\sigma$ 的谱系是所有的正整数 (任意大的非域结构都存在).

1955 年 Gü nter Asser 提出如下问题: 谱系的补集是否还是谱系? 只要看到简单地取否定是不可行的 (如上段所述), 就会看出这个问题不是轻易就能回答的. 事实上, 作为著名的谱系问题, 该问题还没有解决. 但是现在的工作已经转移到另外一个公开的问题上, 即是否 co-NP=NP.

17. (a) 考虑带有等号的语言, 除 $\forall$ 外其唯一的参数是二元谓词符号 P. 证明: 如果 $\mathfrak{A}$ 是有限的且 $\mathfrak{A}\equiv\mathfrak{B}$, 那么 $\mathfrak{A}$ 与 $\mathfrak{B}$ 同构. 提示: 假设 $\mathfrak{A}$ 的论域大小为 n. 令单句子 σ 具有 $\exists v_1\cdots\exists v_n\theta$ 的形式, 表示 $\mathfrak{A}$ 是 "完全的" . 即, 一方面, σ 在 $\mathfrak{A}$ 中必定是真的; 另一方面, σ 的任何模型一定与 $\mathfrak{A}$ 同构.

101

*(b) 证明无论语言中包含什么参数，(a) 的结果都是成立的.

18. 全称公式 ($\forall_1$) 是形式为 $\forall x_1 \cdots x_n \theta$ 的公式，存在公式 ($\exists_1$) 是形式为 $\exists x_1 \cdots x_n \theta$ 的公式，其中 θ 是无量词. 设 $\mathfrak{A}$ 是 $\mathfrak{B}$ 的子结构，$s : V \to |\mathfrak{A}|$.

 (a) 证明：如果 $\models_{\mathfrak{A}} \psi[s]$ 且 ψ 是存在公式，那么 $\models_{\mathfrak{B}} \psi[s]$. 如果 $\models_{\mathfrak{B}} \varphi[s]$ 且 φ 是全称公式，那么 $\models_{\mathfrak{A}} \varphi[s]$.

 (b) 推断：句子 $\exists x Px$ 不会等价于任何全称公式，$\forall x Px$ 也不会等价于任何存在公式.

 说明：(a) 的意思是 (当 φ 是句子时) 任何全称句子是"在子结构中都可以保持的". 由于全称是语法性质，其必定与符号串相关. 反过来，在子结构中保持则是语义性质，其必定与结构相关. 但是这一语义性质保持了逻辑等价的语法性质 (这正是我们所需要的). 即，如果 σ 在子结构中总是成立的句子，那么 σ 逻辑等价于全称句子. (这一点要归功于洛斯和塔斯基.)

19. $\exists_2$ 公式是形式为 $\exists x_1 \cdots x_n \theta$ 的公式，其中 θ 是全称公式.

 (a) 证明：如果某语言中不含函数符号 (也不含常量符号)，其中一个 $\exists_2$ 句子在 $\mathfrak{A}$ 中是真的，那么它必定在 $\mathfrak{A}$ 的某个有限子结构中也是真的.

 (b) 推断：$\forall x \exists y Pxy$ 不会与任何 $\exists_2$ 句子等价.

20. 设语言有等号和二元谓词符号 P. 考虑两个结构 $(\mathbb{N}; <)$ 和 $(\mathbb{R}; <)$：

 (a) 试找出一个句子，它在一个结构中是真的，在另外一个中是假的.

 *(b) 证明：任何 $\exists_2$ 句子 (如上题) 在结构 $(\mathbb{N}; <)$ 中为真则必在 $(\mathbb{R}; <)$ 中也为真. 提示：首先，对于任意有限实数的集合，存在 $(\mathbb{R}; <)$ 的从实数到自然数的自同态. 其次，由习题 18，全称公式在子结构中是成立的.

21. 考虑在我们的语言中加入新的量词. 公式 $\exists! x\alpha$(读作存在唯一的 x 满足) 在 $\mathfrak{A}$ 中以 s 满足成立当且仅当存在唯一的一个 $a \in |\mathfrak{A}|$ 使得 $\models_{\mathfrak{A}} \alpha[s(x|a)]$. 设该语言具有相等符号，证明加入的这一新的量词是没有意义的，即原语言中可以找到一个与 $\exists! x\alpha$ 等价的公式.

102

22. 设 $\mathfrak{A}$ 是结构且 h 是函数，$\operatorname{ran} h = |\mathfrak{A}|$. 证明存在结构 $\mathfrak{B}$，使得 h 是一个从 $\mathfrak{B}$ 到 $\mathfrak{A}$ 上的同态. 提示：取 $|\mathfrak{B}| = \operatorname{dom} h$. 一般地，需要使用选择公理在 $\mathfrak{B}$ 中定义函数，除非 h 是一对一的.

 说明：该结果产生"不含等号的升洛文海－斯科伦定理"(见习题 2.6). 即，不含等号时，任意一个结构 $\mathfrak{A}$ 都有任意更高基数的扩充结构 $\mathfrak{B}$，使得 $\mathfrak{A}$ 与 $\mathfrak{B}$ 是初等等价的，除了相等. 除非加入等号，否则这个结论是最强的.

23. 设 $\mathfrak{A}$ 是一个结构，且 g 是一对一函数，定义域为 $|\mathfrak{A}|$. 证明存在唯一的结构 $\mathfrak{B}$ 使得 g 是从 $\mathfrak{A}$ 到 $\mathfrak{B}$ 上的同构.

24. 设 h 是 $\mathfrak{A}$ 到 $\mathfrak{B}$ 中的同构嵌入. 证明存在一个同构于 $\mathfrak{B}$ 的结构 $\mathfrak{C}$，使得 $\mathfrak{A}$ 是 $\mathfrak{C}$ 的子结构. 提示：设 g 是定义域为 $|\mathfrak{B}|$ 的一对一函数，使得 $g(h(a)) = a$，$a \in |\mathfrak{A}|$. 构造 $\mathfrak{C}$ 使得 g 是从 $\mathfrak{B}$ 到 $\mathfrak{C}$ 上的同构.

 说明：该结果并不奇怪. 相反，这是一个明显的结果. 即如果 $\mathfrak{A}$ 可以同构嵌入到 $\mathfrak{B}$ 中，那么就可以将 $\mathfrak{A}$ 看作 $\mathfrak{B}$ 的子结构.

25. 考虑固定的结构 $\mathfrak{A}$. 在语言中，对每个 $a \in |\mathfrak{A}|$，添加一个新的常数符号 c_a. 设 $\mathfrak{A}^+$ 是扩充后的语言的结构，其保持 $\mathfrak{A}$ 的原有参数，并将 c_a 指派给 a. $|\mathfrak{A}|$ 上的关系 R 称为在 A 中可以*由点定义的*当且仅当 R 在 $\mathfrak{A}^+$ 中是可定义的. (与通常的可定义唯一的不同是语言中对 $|\mathfrak{A}|$ 中的元素都有参数.) 设 $\mathfrak{R} = (\mathbb{R}; <, +, \cdot)$.

 (a) 证明：如果 $\mathfrak{A}$ 是 $\mathfrak{R}$ 的子集，包含有限多个区间的并，那么 $\mathfrak{A}$ 在 $\mathfrak{R}$ 中可由点定义.

 (b) 设 $\mathfrak{A} \equiv \mathfrak{R}$，证明 $|\mathfrak{A}|$ 的所有非空、有限，且在 $\mathfrak{A}$ 中可由点定义的子集，在 $|\mathfrak{A}|$ 中有最小上界 (以 $<^{\mathfrak{A}}$ 排序).

 说明：在一个结构中的可定义的标准的说法是"由参数可定义的"；此处使用"点 (point)"是因为"参数"一词在本章中另有其他的含义. 有序实数域可以看成是完备序域. (这一点在分析课中都会讲到.) 但是完备性 (即非空有限集合存在最小上界) 不是一阶性质. 见 4.1 节习题 4. 完备性的一阶

103

"像 (image)" 可以在二阶命题中将 X 替换为一阶公式 φ 得到. 这个结果 (即将 φ 作变元并取全称闭包得到的句子集合) 说明对于可由点定义的集合保持最小上界的性质. 满足这些句子的有序实域称为"实封闭的有序域".

令人惊讶的是这些域不是逻辑学家发明的, 而是代数学家首先研究的, 读者可以在范德瓦尔登 (van der Waerden) 的《近世代数》(*Modern Algebra*) 的第 1 卷中找到相关的内容. 当然, 作者没有涉及逻辑的内容. 塔斯基证明了任意的实闭有序域初等等价于实数域. 由此可知, 实闭有序域是可定义的.

26. (a) 考虑某个结构 $\mathfrak{A}$, 定义其*初等类型*(elementary type) 为初等等价于 $\mathfrak{A}$ 的结构类. 证明: 这个类就是 EC_Δ. 提示: 证明其为 Mod Th $\mathfrak{A}$.
   (b) 称结构类 $\mathcal{K}$ 为*初等封闭的类*或者 ECL, 如果一个结构属于 $\mathcal{K}$, 那么所有与它初等等价的结构都属于 $\mathcal{K}$. 证明任意这样的类是 EC_Δ 类的并. (如果一个类是 EC_Δ 类的并则称为$\mathrm{EC}_{\Delta\Sigma}$类, 这种记法源于拓扑学.)
   (c) 反之, 证明任意 EC_Δ 类的并形成的类是初等等价封闭的.
27. 设语言的参数是 $\forall$ 和二元谓词符号 P, 试列出所有不同构的大小为 2 的结构. 即给出一个结构列表 (其每个结构的域的大小是 2), 使得任意大小为 2 的结构恰好同构于其中的某一个.
28. 对如下每一对结构, 通过给出一个句子使得其在一个结构中是真的, 而在另外一个结构中是假的来证明它们不是初等等价的. (这里的语言包括 $\forall$ 和二元函数符号.)
   (a) $(\mathbb{R}; \times)$ 和 $(\mathbb{R}^*; \times^*)$, 其中 $\times$ 是通常的实数乘法运算, $\mathbb{R}^*$ 是非零的实数集, $\times^*$ 是将 $\times$ 限制在非零实数上的运算.
   (b) $(\mathbb{N}; +)$ 和 $(\mathbb{P}; +^*)$, 其中 $\mathbb{P}$ 是正整数集, $+^*$ 是将通常的加法 $+$ 限制在 $\mathbb{P}$ 上的运算.
   (c) 对于 (a), (b) 的 4 个结构中的每一个, 给出一个句子使得其在该结构中是真的, 而在另外 3 个中是假的.

104

2.3 解析算法[1]

如同命题逻辑一样, 我们需要知道公式的唯一分解性, 以找到其构造方式. 这种唯一性对于判断那些使用递归的定义是必要的, 比如上一节中的满足性的定义.

我们使用波兰记法来表示项; 公式要使用括号. 相应地, 首先考虑项的分解过程, 证明唯一可读性. 然后, 将此方法扩充到公式.

项是通过符号函数对变量和常数的运算得到的. 定义符号函数 K, 使得对符号 s, $K(s) = 1 - n$, 其中 n 是项的个数.

$$K(x) = 1 - 0 = 1, \text{ 对变量 } x;$$
$$K(c) = 1 - 0 = 1, \text{ 对常数符号 } c;$$
$$K(f) = 1 - n, \text{ 对 } n \text{ 元函数符号 } f.$$

通过如下方式将 K 扩充到表达式的集合

$$K(s_1 s_2 \cdots s_n) = K(s_1) + K(s_2) + \cdots + K(s_n).$$

由于任何符号都不是其他符号的有限序列, 因此这个定义是无二义性的.

引理 23A 对于任意项 t, $K(t) = 1$.

1. 如果读者愿意接受用递归的方式给出的定义, 本节的内容可以跳过.

证明 对 t 使用归纳法. 对于 n 元函数符号 f 的归纳步骤为

$$K(ft_1\cdots t_n)=(1-n)+\underbrace{(1+\cdots+1)}_{n\text{ 个}}=1 \quad \blacksquare$$

事实上, K 是满足引理 23A 的唯一的符号函数. 由此引理, 如果 ε 是 m 个项的连接, 那么 $K(\varepsilon)=m$.

符号串 $\langle s_1,\cdots,s_n\rangle$ 的 *终段*(terminal segment) 是指形式为 $\langle s_k,s_{k+1},\cdots,s_n\rangle$ 的符号串, 其中 $1\leqslant k\leqslant n$.

105

引理 23B 项的任意终段是一个或者多个项的连接.

证明 对项使用归纳法. 对一个符号的项 (即一个变量或者一个常数符号), 结论是显然的. 对项 $ft_1,\cdots,t_n$, 其任意终段 (其自身除外) 必定等于

$$t'_k t_{k+1}\cdots t_n,$$

其中 $k\leqslant n$ 且 t'_k 是 t_k 的终段. 由归纳假设, t'_k 是 m 个项的连接, 其中 $m\geqslant 1$. 这样, 就有 $m+(n-k)$ 项. ■

推论 23C 一个项的任何真的初始段都不是项. 如果 t_1 是 t 的真的初始段, 那么 $K(t_1)<1$.

证明 设项 t 被分为真的初始段 t_1 和终段 t_2, 那么 $1=K(t)=K(t_1)+K(t_2)$, 且由引理 23B, $K(t_2)\geqslant 1$. 因此, $K(t_1)<1$, 且 t_1 不是项. ■

2.3.1 项的解析

对于给定的表达式, 需要一个算法判定是不是合法的项, 如果是, 则要构建其唯一的树, 以说明这个项是如何构成的.

对于给定的一个表达式, 我们要构建一棵树, 树根就是给定的表达式. 开始, 树中只有这一个节点, 但是随着构建过程的继续, 树开始向下生长.

算法包括以下两步:

(1) 如果每个最小的节点 (最底部) 只有一个符号 (必定 [1] 是一个变量或者一个常数符号), 那么算法结束. 否则, 选择一个具有多个符号的最小的节点, 检查其表达式.

(2) 第一个符号必定 [1] 是 n 元函数符号, 记作 f, 这里的 $n>0$. 在当前节点下创建 n 个新的节点. 扫描 f 后面的表达式, 直到遇到第一个 $K(t)=1$ 的 (关于变量, 常数符号和函数符号的) 串 t, 那么 t 就是最左边的未标记的新节点的表达式. 对表达式的其余部分重复此过程, 直到所有 n 个新节点都被标记, 该表达式全部完成 [2], 返回第 1 步.

106

与 1.3 节类似, 关键问题在于树要能够唯一确定. 在第 2 步, 选择第一个满足 $K(t)=1$ 的串 t. 不能使用比 t 更小的 (使用引理 2.3A, 就要求 $K(t)=1$), 也不能使用比 t 更长的 (因为更长的串含有 $K(t)=1$ 的真的初始段, 与推论 23C 矛盾), 这样 t 的选择就是唯一的.

如果算法终止, 那么结构可能有两个: (1) 给定的表达式不是项; (2) 已经构建成了一棵树, 该树能够表示表达式, 且是合法构建的.

1. 否则, 此处的表达式不是项. 我们拒绝给定的表达式不是项, 停止.

2. 如果在找到这样的 t 之前到达表达式的结尾, 那么这样的表达式不是项, 我们拒绝给定的表达式不是项, 停止.

我们可以强调 1.4 节中的唯一性如下.

项的唯一可解释性定理 项集是由变量集和常数符号通过 $\mathcal{F}_f$ 运算自由生成的.

证明 首先, 显而易见, 如果 $f \neq g$, 那么 ran $\mathcal{F}_f$ 与 $\mathcal{F}_g$ 是不交的; 这只需要检查第一个符号. 其次, 来自变量和常数符号的集合的两个值域也是不交的. 只需证明, 当限制到项上时, $\mathcal{F}_f$ 是一对一的. 假设对二元函数 f, 有

$$ft_1t_2 = ft_3t_4.$$

删除第一个字符后得到

$$t_1t_2 = t_3t_4.$$

如果 $t_1 \neq t_3$, 那么其中一个就是另外一个的真初始段, 由推论 23C, 这是不可能的. 因此 $t_1 = t_3$, 接下来, 类似地可以得到 $t_2 = t_4$.

2.3.2 公式的解析

将上述方法扩充到公式上, 现在定义 K 在其他符号上的取值:

$$\begin{aligned}
&K(()=-1;\\
&K())=1;\\
&K(\forall)=-1;\\
&K(\neg)=0;\\
&K(\to)=-1;\\
&K(P)=1-n \quad \text{对 } n \text{ 元谓词符号 } P;\\
&K(=)=-1.
\end{aligned}$$

其实定义的基本思想还是 $K(s)$ 的取值为 $1-n$, 其中 n 是对象的数目 (右括号, 项或者公式). 将 K 扩充到所有表达式的集合上, 即

107

$$K(s_1\cdots s_n) = K(s_1)+\cdots+K(s_n).$$

引理 23D 对于任意合式公式 α, $K(\alpha)=1$.

证明 直接使用归纳法即得. ■

引理 23E 对于合式公式 α 的任意真的初始段 α', $K(\alpha')<1$.

证明 对 α 使用归纳法, 详细步骤留作习题 (习题 1). ■

推论 23F 公式的任何真的初始段都不是公式.

有了这些以后, 就可以像 1.3 节那样进一步深入学习了. 将最小的节点的命题符号替换为原子公式 (区别: 原来是 n 项, 现在则换成了 n 元谓词符号).

非原子的合式公式的开始有两种: $\forall v_i$ 或者 (. 前一种情况, 需要建立新的节点; 而后一种情况, 则需要检查下一个符号是不是 $\neg$. 如果不是, 则需要对括号计数或者使用 K 函数——这两种方法都可以使用——以进行正确分割.

再次，与 1.4 节类似的唯一可解释性如下：

公式的唯一可解释性定理　合式公式的集合是由原子公式通过 $\mathcal{E}_{\neg}, \mathcal{E}_{\rightarrow}$ 和 $Q_i(i=1,2,\cdots)$ 运算自由生成的.

证明　一元运算 $\mathcal{E}_{\neg}$ 和 Q_i 显然都是一对一的，与 1.4 节中一样，可以证明限制到合式公式上是一对一的.

基于如下不同之处，定理证明的另一部分是不同的.

(1) 对于 $i \neq j$, ran$\mathcal{E}_{\neg}$, ran Q_i, ran Q_j 和原子公式的集合是互不相交的. (只需要看前两个符号.)

(2) 类似地，对于 $i \neq j$, ran$\mathcal{E}_{\rightarrow}$, ran Q_i, ran Q_j 和原子公式的集合也是互不相交的.

(3) 对于合式公式 β, $(\neg\ \alpha) \neq (\beta \rightarrow \gamma)$，因为任何合式公式的起始符号都不是 $\neg$，因此，$\mathcal{E}_{\neg}$ 与 $\mathcal{E}_{\rightarrow}$ 限制到合式公式的取值是不交的.

习题

1. 证明对于一个合式公式 α 的真的初始段 α', $K(\alpha') < 1$.

108

2. 设 ε 是包含变量，常数符号和函数符号的表达式. 证明：ε 是项当且仅当 $K(\varepsilon)=1$且对 ε 的任意终段 ε'，我们有 $K(\varepsilon')>0$. 提示：证明更强一些的结果：如果对 ε 的任意终段 ε', $K(\varepsilon')>0$，那么 ε 是 $K(\varepsilon)$ 个项的连接. (这一算法要归功于 Jáskowski.)

2.4　演绎计算

假如有 $\Sigma \vDash \tau$，我们应该用什么方法来证明它呢？这需要进行证明吗？

这个问题需要考虑证明是如何构成的. 证明就是给出论据，使得人们能够彻底相信我们给出的断言 (在这里，就是 $\Sigma \vDash \tau$) 是正确的.

这就要求证明应该是有限长的，因为我们无法给人以无限长的东西. 如果假设条件的集合 Σ 是无限的，那么我们也只能使用其中有限的一部分. 一阶逻辑的紧致性定理可以保证存在有限集合 $\Sigma_0 \subseteq \Sigma$，其中 Σ_0 满足 $\Sigma_0 \vDash \tau$. (在 2.5 节，将使用本节的演绎运算证明该定理.)

除长度要求有限外，证明的另外一个本质特征是能够进行验证，以确保其不包含错误. 这个验证必须是恒真的；是可以按照步骤实施的，而不能仅凭验证者的灵感闪现. 特别地，从无假设条件开始的证明 (即对 $\vDash \tau$ 的证明) 的集合必须是可判定的. 这意味着无假设条件的可证明的公式集合必须是能行可枚举的. 这是因为，原则上，通过生成所有符号串和从中将可证明的句子找出来的方式，可以枚举出所有可证明的句子. 当找到一个证明时，其最后一行就出现在可证明列表中. (在 2.5 节的结尾，有更详细的论述.) 此处的可枚举定理 (在 2.5 节证明) 说明：在合理的条件下，恒真的公式集确实是可以枚举的.

这样，对于逻辑蕴涵满足性的证明的存在性，紧致性定理和可枚举定理就是必要条件. 反过来，这两个定理也是证明存在性的充分条件. 设 $\Sigma \vDash \tau$，由紧致性定理，存在有限集合 $\{\sigma_0,\cdots,\sigma_n\} \subseteq \Sigma$ 逻辑蕴涵 τ，那么 $\tau_0 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \tau$ 就是恒真的 (2.2 节的习题 1). 这样，

109

为证明 $\Sigma \vDash \tau$，就需要执行对恒真公式的有限次枚举，直到出现 $\tau_0 \rightarrow \cdots \rightarrow \sigma_n \rightarrow \tau$ 为止，并对每个 i，验证 $\sigma_i \in \Sigma$. (这可以和 2.2 节讨论的逻辑蕴涵的原始定义的复杂过程进行比较.)

记录枚举的过程，就会产生 $\tau_0 \to \cdots \to \sigma_n \to \tau$，这就可以看作 $\Sigma \vDash \tau$ 的 证明. 作为一个证明，那些认为能行枚举过程是正确的人能够接受这一证明.

与前面的讨论不同，本节主要讨论的问题如下：介绍形式证明，我们称之为 *演绎*，以避免与自然语言中的证明混淆. (在我们的演绎思想模型中) 这将会反映数学家所做的证明，这些证明为了向其同行证实某个特定的事实. 然后在 2.5 节证明，只要 $\Sigma \vDash \tau$，就会存在一个从 Σ 到 τ 的演绎. 如前所述，我们会得到紧致性定理和可枚举定理的证明. 在此过程中，我们会明确演绎方法能够足以说明，给定的句子实际上是由其他命题逻辑蕴涵的. 换句话说，我们的目标是得到一个关于演绎的简洁的数学概念，这个概念在一阶逻辑的环境下是充分而正确的.

2.4.1 形式演绎

选择一个有限的公式集 Λ，称之为逻辑公理；并制定推理规则，这些规则可以使我们从某些公式获得新的公式. 然后对于公式集 Γ，Γ 的 *定理* 是指由 $\Gamma \cup \Lambda$ 中的公式通过 (有限次) 使用推理规则得到的公式. 如果 φ 是 Γ 的定理 (记作：$\Gamma \vdash \varphi$)，那么为了得到公式 φ，由 $\Gamma \cup \Lambda$ 使用推理规则的公式序列称之为从 Γ 到 φ 的一个演绎 (在下面解释).

推理规则的选择当然不止一种. 在本节中，我们给出一阶逻辑的一个演绎计算方法，该计算方法是很多可能方法中的一个. (例如，可以取 $\Lambda = \phi$，通过选择使用一些推理规则. 当然，也可以取另一个极端，选择 Λ 为无限多的公式集，但只使用一条规则.)

我们选择的推理规则是传统上著名的 *假言推理* (modus ponen)，通常表述为：从公式 α 和 $\alpha \to \beta$，可以得到 β：

$$\frac{\alpha, \alpha \to \beta}{\beta}.$$

这样，公式集 Γ 的定理就是从 $\Gamma \cup \Lambda$ 通过有限次使用假言推理可以得到的公式. 110

定义 *从 Γ 到 φ 的一个演绎* 是一个有限的公式序列 $\langle \alpha_0, \cdots, \alpha_n \rangle$，使得 $\alpha_n = \varphi$，且对每个 $k \leqslant n$，或者 (a) α_k 在 $\Gamma \cup \Lambda$ 中；或者 (b) α_k 可以由序列中出现在该公式之前的两个公式通过假言推理得到，即，对小于 k 的 i 和 j，α_j 是 $\alpha_i \to \alpha_k$.

如果存在这样的演绎，则称 φ 是由 Γ*可演绎推出的*，或者 φ 是 Γ 的 *定理*，记作 $\Gamma \vdash \varphi$.

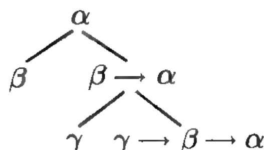
可以从另外一个角度来看这个问题：由 Γ 到 φ 的一个演绎可以看作是一个 *构造序列*，该序列说明 φ 是如何从 $\Gamma \cup \Lambda$ 通过有限次使用假言推理得到的. (我们不能说 φ 是从 $\Gamma \cup \Lambda$ 构建的. 这不同于从短的公式生成长公式的构造运算，因为假言推理可以从长的公式得到短的公式.) 即 Γ 的定理集合恰好是由集合 $\Gamma \cup \wedge$ 通过假言推理可以获得的公式集.

这与 1.4 节讨论的情况有两点不同：一个重要的和一个不重要的. 不重要的区别在于可以通过假言推理“部分定义”的运算来得到定理集，运算的定义域只包括形如 $\langle \alpha, \alpha \to \beta \rangle$ 的公式对 (相对于“全局定义”的构建公式的运算). 更重要的区别则在于定理集不是由 $\Gamma \cup \Lambda$ 通过使用假言推理自由生成的. 这反映了一个事实：一个定理不只有一个唯一的演绎. 在 1.3 节和 2.3 节，我们关心的是对于任意的合式公式存在唯一的树 (这可以恒真计算出来)，用以说明 φ 是如何使用公式构造运算来建立的. 这里，树不再是唯一的，其计算方式也就大不相同了.

尽管如此，还是可以得到如下的归纳原理．只要 $\alpha \in S$ 且 $(\alpha \to \beta) \in S$，那么 $\beta \in S$，则称公式集 S 在假言推理下是封闭的．

归纳原理 设 S 是包含 $\Gamma \cup \Lambda$ 的合式公式的集合，且在假言推理下是封闭的，那么 S 包含 Γ 的所有定理．

例如，如果公式 β，γ 和 $\gamma \to \beta \to \alpha$ 都在 $\Gamma \cup \Lambda$ 中，那么 $\Gamma \vdash \alpha$，下面的树说明 α 是如何得到的，也说明了 $\Gamma \vdash \alpha$.



[111] 尽管可以将演绎定义为这样的树，但是将其看作由这样的树压缩成的线性序列会显得更简单一些．

现在，我们给出逻辑公理的集合 Λ，可以将其分为 6 组．称合式公式 φ 是 ψ 的概化(generalization) 当且仅当对某个 $n \geqslant 0$ 和某些变量 $x_1, \cdots, x_n$，

$$\varphi = \forall x_1 \cdots \forall x_n \psi.$$

这里包含 $n = 0$ 的情形，即任意合式公式都是其自身的一个概化．逻辑公理都是如下形式合式公式的概化，其中 x 和 y 都是变量，并且 α 和 β 是合式公式．

(1) 重言式．

(2) $\forall x\alpha \to \alpha_t^x$，其中 t 是 x 在 α 中的替换．

(3) $\forall x(\alpha \to \beta) \to (\forall x\alpha \to \forall x\beta)$.

(4) $\alpha \to \forall x\alpha$，其中 x 在 α 中不是自由出现的．

如果语言包含相等符号，那么需要加上如下两组：

(5) $x = x$;

(6) $x = y \to (\alpha \to \alpha')$，其中 α 是原子的且 α' 是有限次的将 α 中的 x 替换为 y 得到的．

第 3~6 组公理的大部分都是清楚的，后面会有很多例子．第 1 和 2 组公理需要解释一下．当然，上述 6 组逻辑公理的出现有些突兀，后面将会对其中每一组的来源进行说明．

2.4.2 替换

在第 2 组公理中，我们发现

$$\forall x\alpha \to \alpha_t^x.$$

这里 α_t^x 是将公式 α 中自由出现的变量 x 都替换为 t 所得到的表达式．这个概念也可以通过递归进行定义：

(1) 对原子公式 α，α_t^x 是将公式 α 中的所有变量 x 替换为 t 所得到的表达式．(详见习题 1，注意 α_t^x 本身是公式．)

(2) $(\neg\, \alpha)_t^x = (\neg\, \alpha_t^x)$.

(3) $(\alpha \to \beta)_t^x = (\alpha_t^x \to \beta_t^x)$.

(4) $(\forall y\alpha)_t^x = \begin{cases} \forall y\alpha & \text{如果 } x = y, \\ \forall y(\alpha_t^x) & \text{如果 } x \neq y. \end{cases}$

112

例 (1) $\varphi_x^x = \varphi$.

(2) $(Qx \to \forall xPx)_y^x = (Qy \to \forall xPx)$.

(3) 如果 α 是 $\neg\forall yx = y$, 那么 $\forall x\alpha \to \alpha_z^x$ 是

$$\forall x \neg \forall yx = y \to \neg \forall yz = y.$$

(4) 对 (3) 中的 α, $\forall x\alpha \to \alpha_y^x$ 是

$$\forall x \neg \forall yx = y \to \neg \forall yy = y.$$

上述最后的一个例子说明了必须要注意的问题. 总体上说, $\forall x\alpha \to \alpha_t^x$ 似乎是一个非常合理的公理. (“如果 α 对于任意对象都是真的, 那么对于 t 也肯定是真的.”) 但是在例 4 中, 形式为 $\forall x\alpha \to \alpha_t^x$ 的句子几乎总是假的. 其前提 $\forall x\neg\forall yx = y$ 在论域里有两个或者多个元素的任何结构下都是真的. 但是其结论 $\neg\forall yy = y$ 在任何结构下都是假的. 因此, 该句子就是错误的.

问题是当 y 替换 x 后, 受到了全称量词 $\forall y$ 的限制. 需要对第 2 组公理加以限制使之能够排除这类量词的限制问题. 非正式地, 如果在项 t 中有某个 y 在 α_t^x 中受到全称量词 $\forall y$ 的限制, 则项 t 不能够替换 α 中的 x. 其正式定义可以使用递归定义如下. (因为后面的归纳证明中会用到这个概念, 所以递归定义实际上是最有用的.)

设 x 是变量, t 是项, 定义“在 α 中 t 可以替换 x”如下:

(1) 对原子公式 α, 在 α 中 t 可以替换 x. (在 α 中没有量词, 因此也不会受其限制.)

(2) 在 $(\neg\alpha)$ 中 t 可以替换 x 当且仅当在 α 中 t 可以替换 x. 在 $(\alpha \to \beta)$ 中 t 可以替换 x 当且仅当在 α 和 β 中 t 都可以替换 x.

(3) 在 $\forall y\alpha$ 中 t 可以替换 x 当且仅当或者 (a) x 在 $\forall y\alpha$ 中不是自由出现的, 或者 (b) y 在 t 中不出现且在 α 中 t 可以替换 x.

(这一点是为了保证在 t 中没有受到前缀 $\forall y$ 限制的对象, 且在 α 内部没有出现问题.)

例如, x 在任何公式中总可以替换其自己. 如果 t 不包含出现在 α 中的变量, 那么在 α 中 t 可以替换 x.

读者要注意不要混淆这些词. 即使 t 不能够替换 α 中的 x, 由 α 通过把那些自由出现的 x 替换为 t, 仍可得到 α_t^x. 用这样的替换来构造 α_t^x, 即使非常谨慎的人也不会认为此为明智之举.

113

第 2 组公理包括了具有如下形式的公式的所有概化:

$$\forall x\alpha \to \alpha_t^x,$$

其中项 t 在公式 α 中可以替换 x. 例如, 在第 2 组公理中

$$\forall v_3(\forall v_1(Av_1 \to \forall v_2 Av_2) \to (Av_2 \to \forall v_2 Av_2))$$

其中 x 是 v_1, α 是 $Av_1 \to \forall v_2 Av_2$, 且 t 是 v_2. 另一方面,

$$\forall v_1 \forall v_2 Bv_1 v_2 \to \forall v_2 Bv_2 v_2$$

不在第 2 组公理中，因为 v_2 在 $\forall v_2 Bv_1v_2$ 中不可替换 v_1.

2.4.3　重言式

第 1 组公理包含了对所有重言式的概化. 这些合式公式能够从命题逻辑的重言式 (命题逻辑中只有 $\neg$ 和 $\to$ 两个联结词) 通过使用一阶语言的合式公式替换每个句子而获得. 例如，

$$\forall x[(\forall y\neg\, Py \to \neg\, Px) \to (Px \to \neg\, \forall y\neg\, Py)]$$

属于第 1 组公理，是方括号中的公式的概化，而方括号中的公式是由如下逆否重言式通过将 $\mathbf{A}$, $\mathbf{B}$ 分别替换为 $\forall y\neg\, Py$ 和 Px 而得到的：

$$(\mathbf{A} \to \neg\, \mathbf{B}) \to (\mathbf{B} \to \neg\, \mathbf{A})$$

下面介绍另外一种更直接的看待第 1 组公理的方式. 将合式公式分为两类.

(1) 基本(prime) 公式是原子公式和形式为 $\forall x\alpha$ 的公式；

(2) 其他的是非基本公式，即形式为 $\neg\,\alpha$ 和 $\alpha \to \beta$ 的公式.

这样，任何公式都可以由基本公式通过 $\varepsilon_\neg$ 和 $\varepsilon_\to$ 的运算构造. 回过头来看命题逻辑，将命题符号看作一阶语言的基本公式. 那么命题逻辑的重言式 (只有 $\neg$ 和 $\to$ 两个联结词) 属于第 1 组的公理. 无需将命题符号替换为一阶合式公式，因为它们已经是一阶合式公式了. 反之，第 1 组公理都是命题逻辑的重言式的概化. (其证明使用 1.2 节的习题 8.)

114

例　$(\forall y\neg\, Py \to \neg\, Px) \to (Px \to \neg\, \forall y\neg\, Py)$.

这个公式包含了两个命题符号 (基本公式)，$\forall y\neg\, Py$ 和 Px. 其真值表如下：

$$\begin{array}{cccccccc}
(\forall y\neg\, Py & \to & \neg & Px) & \to & (Px & \to & \neg\, \forall y\neg\, Py) \\
T & F & F & T & \mathbf{T} & T & F & F\,T \\
T & T & T & F & \mathbf{T} & F & T & F\,T \\
F & T & F & T & \mathbf{T} & T & T & T\,F \\
F & T & T & F & \mathbf{T} & F & T & T\,F
\end{array}$$

由此表可以看出，这的确是一个重言式.

另一方面，$\forall x(Px \to Px)$ 和 $\forall xPx \to Px$ 都不是重言式.

说明 1：这里我们没有假定一阶语言只有可数多个公式. 因此，实际上是将第 1 章推广到了不可数的符号集的情形.

说明 2：取所有重言式作为逻辑公理是太多了. 可以取其中很少的一部分，这样做的代价是演绎推理的长度会增加. 一方面，重言式构成了一个良好的可判定集合 (其可判定性对于 2.5 节的可枚举定理是非常重要的.) 另一方面，正如 1.7 节所述，重言式没有很好的判定方法. 一个选择是将第 1 组公理削减为部分重言式的集合，这部分重言式的判定方法是快速的 (使用术语应该说是 “多项式时间可判定的”). 其他的重言式可以由此在假言推理的作用下得到.

说明 3：即使一阶公式也是命题逻辑的合式公式，对其仍可以使用第 1 章和第 2 章中的概念. 如果 Γ 重言蕴涵 φ，那么 Γ 也逻辑蕴涵 φ(见习题 3). 但是其反过来是不成立的. 例如，$\forall xPx$ 逻辑蕴涵 Pc，但是 $\forall xPx$ 不会重言蕴涵 Pc，因为这是两个不同的命题符号.

定理 24B $\Gamma \vdash \varphi$ 当且仅当 $\Gamma \cup \Lambda$ 重言蕴涵 φ.

证明 ($\Rightarrow$)：这依赖于一个明显的事实：$\{\alpha, \alpha \to \beta\}$ 重言蕴涵 β. 设有真值指派 v 满足 $\Gamma \cup \Lambda$ 的每个元素. 由归纳法可以看出，v 满足 Γ 的每个定理. 归纳步骤恰好使用了上述事实.

($\Leftarrow$)：设 $\Gamma \cup \Lambda$ 重言蕴涵 φ. 那么由紧致性定理 (命题逻辑) 的推论，存在有限集合 $\{\gamma_1, \cdots, \gamma_m, \lambda_1, \cdots, \lambda_n\}$ 重言蕴涵 φ. 那么，

$$\gamma_1 \to \cdots \to \gamma_m \to \lambda_1 \to \cdots \lambda_n \to \varphi$$

115

是重言式 (见 1.2 节的习题 4)，因此也在 Λ 中. 对该重言式和 $\{\gamma_1, \cdots, \gamma_m, \lambda_1, \cdots, \lambda_n\}$ 使用 $m+n$ 次假言推理，可得到 φ. ■

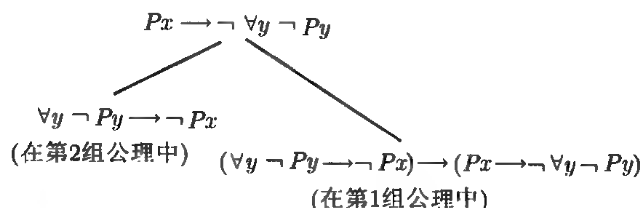
(上述证明与 1.7 节的习题 7 有关，用到了对有可能是不可数语言的命题的紧致性.)

2.4.4 演绎与元定理

我们已经完成对逻辑公理集 Λ 的介绍，集合 Γ 的定理集是由 $\Gamma \cup \Lambda$ 通过假言推理建立起来的. 例如，

$$\vdash Px \to \exists y Py.$$

(这里 $\Gamma = \varnothing$，记作"$\vdash \alpha$"而不是"$\varnothing \vdash \alpha$".) 公式 $Px \to \exists y Py$ 可以通过对 Λ 的两个元素使用假言推理得到，如下面的家谱树所示：

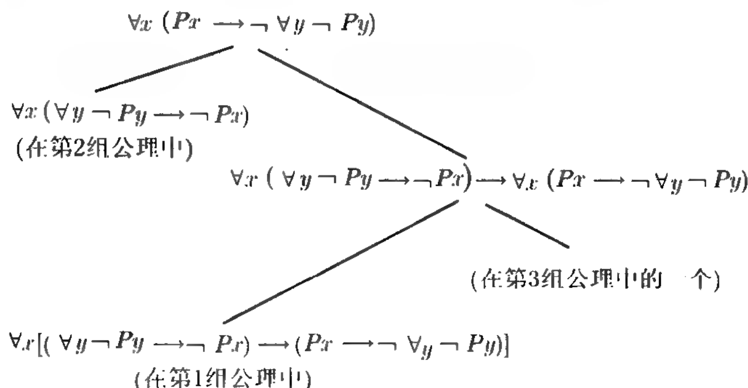


将此树压缩成线性的三元序列，我们就得到 (从 $\varnothing$ 开始的) 公式 $Px \to \exists y Py$ 的一个演绎.

作为另外一个例子，可以得到该公式的一个概化：

$$\vdash \forall x(Px \to \exists y Py).$$

这可由如下的树得到，这棵树显示了由 Λ 使用假言推理得到 $\forall x(Px \to \exists y Py)$ 的构造：



116

再次将这棵树压缩成演绎.

在这些例子中，家谱树看上去好像作用不大. 稍后将介绍使用系统的方法生成这样的树. 这些方法很大程度上要依赖于下面的概化定理和演绎定理.

注意：这里定理的含义有两个不同的层次. 如果 $\Gamma \vdash \alpha$，称 α 是 Γ 的定理. 自然语言中也有很多称为定理的陈述. 这两个层次的定理一般不会混淆. 为了强调自然语言中的演绎和定理的结果，可以将自然语言中的定理称为*元定理*.

概化定理反映了非形式的感觉：如果在没有对 x 特定假设的前提下能够证明 $_x_$，那就可以说“因为 x 是任意的，故有 $\forall x_x_$”.

概化定理　如果 $\Gamma \vdash \varphi$ 且 x 不在 Γ 的任何公式中自由出现，那么 $\Gamma \vdash \forall x\varphi$.

证明　考虑给定的公式集 Γ 和不在 Γ 中自由出现的变量 x. 下面使用归纳法证明对 Γ 的任意定理 φ，有 $\Gamma \vdash \forall x\varphi$. (使用归纳原理) 可以证明集合

$$\{\varphi | \Gamma \vdash \forall x\varphi\}$$

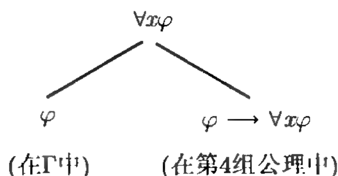
包含 $\Gamma \cup \Lambda$，且在假言推理下是封闭的. 注意 x 在 φ 中能够自由出现.

情形 1：φ 是逻辑公理，那么 $\forall x\varphi$ 也是逻辑公理，因此 $\Gamma \vdash \forall x\varphi$.

情形 2：$\varphi \in \Gamma$. 那么 x 在 φ 中不是自由出现的，因此，

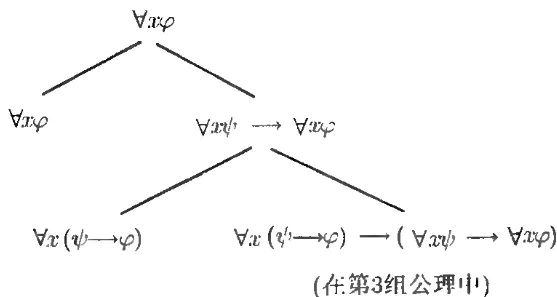
$$\varphi \to \forall x\varphi$$

是第 4 组公理. 这样，由如下的树可以得到 $\Gamma \vdash \forall x\varphi$：



情形 3：φ 是由 ψ 和 $\psi \to \varphi$ 经假言推理得到的. 由归纳假设，可以得到 $\Gamma \vdash \forall x\psi$ 和 $\Gamma \vdash \forall x(\psi \to \varphi)$. 在这种情形下，第 3 组公理是有用的. 那么由如下的树可以得到 $\Gamma \vdash \forall x\varphi$：因此由归纳法，对 Γ 的每个定理 φ，$\Gamma \vdash \forall x\varphi$.

117



(第 3 组和第 4 组公理的理由可由上述证明看出.) ■

“x 不在 Γ 中自由出现”的限制是必要的. 例如，$Px \nvDash \forall xPx$，根据 2.5 节的可靠性定

理，$Px \nvdash \forall x Px$. 另一方面，通常 x 在 φ 中是自由出现的. 例如，本节开始的第 1 个例子

$$\vdash (Px \to \exists y Py).$$

第 2 个例子

$$\vdash \forall x(Px \to \exists y Py),$$

这个例子可以由第 1 个例子根据上述证明的情形 3 得到.

例 $\forall x \forall y \alpha \vdash \forall y \forall x \alpha$.

概化定理的证明事实上比定理本身说明的问题更多. 这个证明说明，一旦给定了从 Γ 到 φ 的演绎，就可以将其转化为从 Γ 到 $\forall x \varphi$ 的演绎.

引理 24C(规则 T) 如果 $\Gamma \vdash \alpha_1, \cdots, \Gamma \vdash \alpha_n$，且 $\{\alpha_1, \cdots, \alpha_n\}$ 重言蕴涵 β，那么 $\Gamma \vdash \beta$.

证明 $\alpha_1 \to \cdots \alpha_n \to \beta$ 是重言式，因此也是一个逻辑公理. 使用 n 次假言推理即得. ■

演绎定理 如果 $\Gamma; \gamma \vdash \varphi$，那么 $\Gamma \vdash (\gamma \to \varphi)$.

(其逆定理很明显也是成立的；事实上，逆定理本质上就是假言推理.)

证明 1 $\Gamma; \gamma \vdash \varphi$ iff $(\Gamma; \gamma) \cup \Lambda$ 重言蕴涵 φ;
iff $\Gamma \cup \Lambda$ 重言蕴涵 $(\gamma \to \varphi)$;
iff $\Gamma \vdash (\gamma \to \varphi)$.

118

证明 2 这个证明不像上面的证明那样使用紧致性定理. 这个证明直接给出如何从 $\Gamma; \gamma$ 到 φ 的演绎转化为从 Γ 到 $(\gamma \to \varphi)$ 的演绎. 我们使用归纳法证明对于 $\Gamma; \gamma$ 的任意定理 φ，公式 $(\gamma \to \varphi)$ 是 Γ 的定理.

情形 1：$\varphi = \gamma$. 显然，$\vdash (\gamma \to \varphi)$.

情形 2：φ 是逻辑公理或者是 Γ 的元素，那么 $\Gamma \vdash \varphi$. 且 φ 重言蕴涵 $(\gamma \to \varphi)$，因此由规则 **T** 知，$\Gamma \vdash (\gamma \to \varphi)$.

情形 3：φ 是由 ψ 和 $\psi \to \varphi$ 经假言推理得到的. 由归纳假设，可以得到 $\Gamma \vdash (\gamma \to \varphi)$ 和 $\Gamma \vdash (\gamma \to (\psi \to \varphi))$. 集合 $\{\gamma \to \psi, \gamma \to (\psi \to \varphi)\}$ 重言蕴涵 $\gamma \to \psi$. 这样，由规则 **T**，$\Gamma \vdash (\gamma \to \varphi)$.

由归纳法，对于任意由 $\Gamma; \gamma$ 演绎的 φ，结论都是成立的. ■

推论 24D(逆否) $\Gamma; \varphi \vdash \neg \psi$ 当且仅当 $\Gamma; \psi \vdash \neg \varphi$.

证明 $\Gamma; \varphi \vdash \neg \psi \Rightarrow \Gamma \vdash \varphi \to \neg \psi$ 由演绎定理,
$\Rightarrow \Gamma \vdash \psi \to \neg \varphi$ 由规则 **T**,
$\Rightarrow \Gamma; \psi \vdash \neg \varphi$ 由假言推理.

(第 2 步中用到了 $\varphi \to \neg \psi$ 重言蕴涵 $\psi \to \neg \varphi$.) 由对称性，反过来也成立. ■

称一个公式集是*不和谐的*(inconsistent) 当且仅当对某个 β，β 与 $\neg \beta$ 都是这个集合的定理. (在这种情形下，任意公式 α 都是该集合的定理，原因在于 $\beta \to \neg \beta \to \alpha$ 是重言式.)

推论 24E(归谬法) $\Gamma; \varphi$ 是不和谐的，那么 $\Gamma \vdash \neg \varphi$.

证明 由演绎定理，有 $\Gamma \vdash (\varphi \to \beta)$ 和 $\Gamma \vdash (\varphi \to \neg\beta)$. $\{\varphi \to \beta, \varphi \to \neg\beta\}$ 重言蕴涵 $\neg\varphi$. ■

例 $\vdash \exists x \forall y \varphi \to \forall y \exists x \varphi$.

由演绎定理，需要证明 $\exists x \forall y \varphi \vdash \forall y \exists x \varphi$.

119

由概化定理，需要证明 $\exists x \forall y \varphi \vdash \exists x \varphi$.

与上式等价的是 $\neg \forall x \neg \forall y \varphi \vdash \neg \forall x \neg \varphi$.

由逆否定理和规则 **T**，需要证明 $\forall x \neg \varphi \vdash \forall x \neg \forall y \varphi$.

由概化定理，需要证明 $\forall x \neg \varphi \vdash \neg \forall y \varphi$.

由归谬法可以证明，$\{\forall x \neg \varphi, \forall y \varphi\}$ 是不和谐的.

容易得到：

(1) 由第 2 组公理和假言推理，有 $\forall x \neg \varphi \vdash \neg \varphi$.

(2) 同样，$\forall y \varphi \vdash \varphi$.

上述两行说明 $\{\forall x \neg \varphi, \forall y \varphi\}$ 是不和谐的.

2.4.5 策略

如前面的例子所示，概化定理和演绎定理 (一定程度上，包括推论) 在证明某个特定的公式是否可演绎时是非常有用的. 但是，这里还存在策略的问题：对于给定的 Γ 与 φ，为证明 $\Gamma \vdash \varphi$，我们该从哪里开始？或许有人会认为，可以枚举所有合式公式的有限序列，直到碰到一个从 Γ 到 φ 的演绎为止. (对于合理的语言而言，) 如果演绎存在，这也许是恒真的；但是，这仅仅是从理论意义上得到的恒真结论.

一种方法是不使用形式语言，而是在自然语言中给出 Γ 的真值蕴涵 φ 的真值的证明. 那么自然语言中的证明可以形式化为形式语言中的合法演绎. (本节后面，将会介绍以合理而自然的方式进行形式化的方法.)

也有一些仅基于 φ 的语法形式的有用方法. 设 φ 是可以由 Γ 演绎的，现在来寻找其证明. 这有几种情况：

(1) 设 φ 是 $(\psi \to \theta)$，那么可以证明 $\Gamma; \psi \vdash \theta$(这总是可能的)；

(2) 设 φ 是 $\forall x \psi$，如果 x 在 Γ 中不是自由出现的，那么可以证明 $\Gamma \vdash \psi$. (即使 x 在 Γ 中自由出现，也可以解决. 总存在变量 y 使得 $\Gamma \vdash y\psi^x_y$，且 $\forall y \psi^x_y \vdash \forall x \psi$. 见习题 9 给出的再替换定理.)

(3) 最后，设 φ 是某个公式的否定.

(3a) 如果 φ 是 $\neg(\psi \to \theta)$，那么可以证明 $\Gamma \vdash \psi$ 和 $\Gamma \vdash \neg\theta$(由规则 **T**)，这总是可能的.

120

(3b) 如果 φ 是 $\neg\neg\psi$，那么肯定可以证明 $\Gamma \vdash \psi$.

(3c) 其他情况 φ 是 $\neg \forall x \psi$，可以证明 $\Gamma \vdash \neg \psi^x_t$，其中 t 是在 ψ 中可替换 x 的某个项. (为什么？) 不幸的是，这并非总是可能的. 有可能会出现如下情形：

$$\Gamma \vdash \neg \forall x \psi,$$

且对每个项 t，

$$\Gamma \nvdash \neg \psi^x_t.$$

(这样的一个例子是 $\Gamma = \varnothing, \psi = \neg (Px \to \forall y Py)$.) 逆否的情形是

$$\Gamma; \alpha \vdash \neg \forall x \psi$$

当且仅当

$$\Gamma; \forall x \psi \vdash \neg \alpha.$$

(一个变形是: $\Gamma; \forall y \alpha \vdash \neg \forall x \psi$ 当 $\Gamma; \forall x \psi \vdash \neg \alpha$ 时) 如果所有情形都不成立, 那么就可以使用归谬法.

例 (Q2A) 如果 x 在 α 中不是自由出现的, 那么

$$\vdash (\alpha \to \forall x \beta) \leftrightarrow \forall x (\alpha \to \beta).$$

这需要证明 (由规则 **T**),

$$\vdash (\alpha \to \forall x \beta) \to \forall x (\alpha \to \beta)$$

和

$$\vdash \forall x (\alpha \to \beta) \to (\alpha \to \forall x \beta).$$

第 1 种情况需要证明 (由演绎定理和概化定理),

$$\{(\alpha \to \forall x \beta), \alpha\} \vdash \beta.$$

这是很容易的, 因为 $\forall x \beta \to \beta$ 是一个公理.

为证明其反面,

$$\vdash \forall x (\alpha \to \beta) \to (\alpha \to \forall x \beta),$$

(由演绎定理和概化定理) 需要证明

$$\{\forall x (\alpha \to \beta), \alpha\} \vdash \beta.$$

这同样是显而易见的.

在上述例子中, 用 $\neg \alpha$ 与 $\neg \beta$ 分别替换 α 和 β, 根据逆否重言式, 可以得到: 121

(Q3B) 如果 x 在 α 中不是自由出现的, 那么

$$\vdash (\exists x \beta \to \alpha) \leftrightarrow \forall x (\beta \to \alpha).$$

读者可自己证明上述公式是恒真的.

如下面的例子所示, 在书写证明时, 经常使用一种缩写的形式.

例 (EQ2) $\forall x \forall y (x = y \to y = x)$.

证明

(1) $\vdash x = y \to x = x \to y = x$. Ax 6.

(2) $\vdash x = x$. Ax 5.

(3) $\vdash x = y \to y = x$.1, 2, ; **T**.

(4) $\vdash \forall x \forall y (x = y \to y = x)$.3; gen^2. ■

第 1 行中, "Ax 6" 表示该公式属于第 6 组公理; 第 3 行 "1.2; **T**" 表示这一行是由第 1 和 2 行根据规则 **T** 得到; 第 4 行中的 "3; gen^2" 表示对第 3 行的公式使用概化定理得到第 4 行. 类似地, 分别用 "MP", "ded" 和 "RAA" 表示假言推理, 演绎定理和归谬法.

需要指出的是上述 4 行并没有构成 $\forall x \forall y (x = y \to y = x)$ 的演绎, 而是给出了演绎存在的证明 (是在元语言中进行的, 也可以说是在自然语言中.) 作者所知道的这个公式最简练的演绎是一个包含 17 个公式的序列.

例 $\vdash x = y \to \forall z Pxz \to \forall z Pyz$.

证明

(1) $\vdash x = y \to Pxz \to Pyz$. Ax 6.

(2) $\vdash \forall z Pxz \to Pxz$. Ax 2.

(3) $\vdash x = y \to \forall z Pxz \to Pyz$.1, 2; T.

(4) $\{x = y, \forall z Pxz\} \vdash Pyz$.3; MP^2.

(5) $\{x = y, \forall z Pxz\} \vdash \forall z Pyz$.4; gen.

(6) $\vdash x = y \to \forall z Pxz \to \forall z Pyz$.5; ded^2. ■

例 (EQ5) 设 f 是二元函数符号, 那么

$$\vdash \forall x_1 \forall x_2 \forall y_1 \forall y_2 (x_1 = y_1 \to x_2 = y_2 \to f x_1 x_2 = f y_1 y_2).$$

证明 第 6 组的两个公理是

$$x_1 = y_1 \to f x_1 x_2 = f x_1 x_2 \to f x_1 x_2 = f y_1 x_2,$$

122

$$x_2 = y_2 \to f x_1 x_2 = f y_1 x_2 \to f x_1 x_2 = f y_1 y_2.$$

从 $\forall x x = x$(在第 5 组公理中), 我们可以推出

$$f x_1 x_2 = f x_1 x_2.$$

上述 3 个公式重言蕴涵

$$x_1 = y_1 \to x_2 = y_2 \to f x_1 x_2 = f y_1 y_2.$$ ■

例(a) $\{\forall x (Px \to Qx), \forall z Pz\} \vdash Qc$. 不难证明其演绎的存在性, 其演绎包含 7 个公式.

(b) $\{\forall x (Px \to Qx), \forall z Pz\} \vdash Qy$. 与 (a) 类似. 这里有意思的是可以使用相同的 7 个公式构成演绎, 只是将其中所有的 c 替换为 y.

(c) $\{\forall x (Px \to Qx), \forall z Pz\} \vdash \forall y Qy$. 由 (b) 及概化定理可得.

(d) $\{\forall x (Px \to Qx), \forall z Pz\} \vdash \forall x Qx$. 由 (c) 及 $\forall y Qy \vdash \forall x Qx$ 这一事实可得.

上述例子中的 (a) 与 (b) 说明常数符号与自由变量的可交换性. 这种可交换性是下面概化定理变形的基础 (c) 是这种变形的一个例子. (d) 包含在推论 2.4G 中. φ_y^c 当然是在 φ 中用 y 替换 c 而得到的结果.

定理 24F(常数的概化) 设 $\Gamma\vdash\varphi$，且 c 是不在 Γ 中出现的常数符号，那么存在变量 y(不出现在 φ 中) 使得 $\Gamma\vdash\forall y\varphi^c_y$. 另外，存在从 Γ 到 $\forall y\varphi^c_y$ 的演绎，在此演绎中不出现 c.

证明 设 $\langle\alpha_1,\cdots,\alpha_n\rangle$ 是从 Γ 到 φ 的演绎，$(\alpha_n=\varphi)$，y 是不出现在任何 α_i 中的第 1 个变量. 称

$$\langle(\alpha_0)^c_y,\cdots,(\alpha_n)^c_y\rangle \qquad (*)$$

是一个从 Γ 到 φ^c_y 的演绎. 因此，我们需要验证每个 $(\alpha_k)^c_y$ 在 $\Gamma\cup\Lambda$ 中或者可以从前面的公式由假言推理得到.

情形 1：$\alpha_k\in\Gamma$. 那么 c 不在 α_k 中出现，因此在 Γ 中，$(\alpha_k)^c_y=\alpha_k$.

情形 2：α_k 是逻辑公理. 那么 $(\alpha_k)^c_y$ 也是逻辑公理. (从逻辑公理列表中，可以看出引入新的变量可以将一个逻辑公理转化为另外一个.)

情形 3：α_k 可以从 α_i 和 α_j(即 $\alpha_i\to\alpha_k$)，i，j 小于 k，由假言推理得到. 那么 $(\alpha_j)^c_y=((\alpha_i)^c_y\to(\alpha_k)^c_y)$. 因此，$(\alpha_k)^c_y$ 可从 $(\alpha_i)^c_y$ 和 $(\alpha_j)^c_y$ 根据假言推理得到.

123

这样就得到 $(*)$ 是 φ^c_y 的演绎的证明. 设 Φ 是 $(*)$ 中应用到的 Γ 的有限子集. 这样 $(*)$ 就是从 Φ 到 φ^c_y 的演绎，且 y 不出现在 Φ 中. 因此根据概化定理，存在从 Φ 到 $\forall y\varphi^c_y$ 的演绎，在此演绎中不出现 c. (概化定理的证明不需要对演绎添加任何新的符号.) 因此，这也是从 Γ 到 $\forall y\varphi^c_y$ 的演绎. ■

有时该定理也用到非任意变量的情况. 在下面的推论中，变量 x 是事先选定的.

推论 24G 设 $\Gamma\vdash\varphi^x_c$，其中常数符号 c 不在 Γ 或者不在 φ 中出现，那么 $\Gamma\vdash\forall x\varphi$，并且存在从 Γ 到 $\forall x\varphi$ 的演绎，Γ 中不出现 c.

证明 由上述定理，存在不出现 c 的从 Γ 到 $\forall y((\varphi^x_c)^c_y)$ 的演绎，其中 y 不出现在 φ^x_c 中，但是由于 c 不出现在 φ 中，所以

$$(\varphi^x_c)^c_y=\varphi^x_y.$$

只需证明 $\forall y\varphi^x_y\vdash\forall x\varphi$. 如果知道

$$(\forall y\varphi^x_y)\to\varphi$$

是一条公理，那么这就很容易了. 也就是说，在 φ^x_y 中 x 可以替换 y，且 $(\varphi^x_y)^y_x$ 必定是 φ. 这是很清楚的，详见习题 9 的再替换引理.

■

推论 24H(EI 规则) 设常数符号 c 不在 φ，ψ 或者 Γ 中出现，且

$$\Gamma;\varphi^x_c\vdash\psi.$$

那么，

$$\Gamma;\exists x\varphi\vdash\psi$$

存在从 $\Gamma;\exists x\varphi$ 到 ψ 的演绎，其中不出现 c.

证明 由逆否律，有

$$\Gamma;\neg\psi\vdash\neg\varphi^x_c.$$

根据前面的推论，有

$$\Gamma;\neg\psi \vdash \forall x \neg\varphi.$$

再次使用逆否律即可. ■

124

"EI" 的含义是"存在实例"，是一个传统术语.

在后面的任何证明中 (习题除外)，我们都不会使用 EI 规则. 其形式化的描述为："已知存在 x 满足 _x_，称之为 c. 由 _c_ 可以证明 ψ. "需要注意 EI 规则并非是指 $\exists x\varphi \vdash \varphi_c^x$，实际上这通常是错误的.

例 $\vdash \exists x\forall y\varphi \to \forall y\exists x\varphi$.

由演绎定理，只需证明

$$\exists x\forall y\varphi \vdash \forall y\exists x\varphi.$$

由 EI 规则，只需证明

$$\forall y\varphi_c^x \vdash \forall y\exists x\varphi,$$

其中 c 是新的符号. 由概化定理，只需证明

$$\forall y\varphi_c^x \vdash \exists x\varphi.$$

因为 $\forall y\varphi_c^x \vdash \varphi_c^x$，只需证明

$$\varphi_c^x \vdash \exists x\varphi.$$

由逆否律，这等价于

$$\forall x\neg\varphi \vdash \neg\varphi_c^x,$$

这是显而易见的 (根据第 2 组公理和假言推理).

现在，我们能够大致地看到逻辑公理列表是如何构成的. 其包含重言式是为了处理命题联结符号. (从这个意义上说，可以考虑只使用其中的一部分重言式.) 第 2 组公理反映了量词符号的含义，为了能够证明概化定理，我们添加了第 3 和 4 组公理.

第 5 组和第 6 组公理能够证明相等的关键性质，见后面关于相等的 2.4.7 节.

正如在 2.5 节要证明的，每个逻辑公理都是恒真的公式. 将逻辑公理看作是所有恒真公式的集合，使用起来或许更简单一些. 但是这样做的目的有两个，一是恒真的概念是根据语义定义的；也就是说定义指的是对于语言来说的可能的含义 (即结构) 以及结构中的真值的概念. 至于现在的目的 (例如，证明恒真集是可枚举的)，我们是需要一个有限的类 Λ 的语法定义. 也就是说，Λ 的定义只关心逻辑公理中的符号排列；不考虑其在结构中的真值问题. 将所有恒真公式都包含在公理中的第二个目的是我们希望有可判定的集合 Λ 和不可判定的恒真集.

125

2.4.6 字母变换式

在考虑如下公式的时候，通常我们对变量 x 和 y 的选择不感兴趣，

$$\forall x(x \neq \mathbf{0} \to \exists y x = \mathbf{S}y)$$

我们希望 $\langle x,y\rangle$ 是一对不同变量，而对 $\langle v_4,v_9\rangle$ 和 $\langle v_8,v_1\rangle$ 通常不加区分.

在用某个项 t 替换到一个公式中去时，约束变量的选择对于 t 是否是可替换的会有影响. 在这一小节，我们讨论如何处理不可替换的情况. 在下面的讨论中，困难总是集中在量化变量上.

例如，我们要证明

$$\vdash \forall x\forall yPxy \to \forall yPyy.$$

其困难是在 $\forall yPxy$ 中. y 不能替换 x，因此，上述句子不在第 2 组公理中，这是由于不慎的变量选择带来的麻烦. 又如，证明

$$\vdash \forall x\forall zPxz \to \forall yPyy.$$

就没有这么麻烦了. 因此，如果知道

$$\vdash \forall x\forall yPxy \to \forall x\forall zPxz,$$

就可以解决问题了，这里同样没有上述麻烦.

对于解决这类特定的问题 (在 $\forall x\forall yPxy$ 和 $\forall yPyy$ 之间插入 $\forall x\forall zPxz$)，这种迂回的方法是很典型的. 也就是说，当我们想要在合式公式 φ 中用项 t 替换 x，在出现不可替换的情况时，我们就用 $\forall x\varphi'$ 替换 $\forall x\varphi$，其中在 φ' 中用项 t 可以替换 x. 在上面的例子中，φ 是 $\forall yPxy$，φ' 是 $\forall zPxz$. 通常，φ 与 φ' 的区别仅在于约束变量的选择. 同时，φ' 的构造必须是恰当的，必须保证与 φ 逻辑等价. 例如，用 $\forall xPxx$ 替换 $\forall yPxy$ 或者用 $\forall z\forall zPxzz$ 替换 $\forall y\forall zQxyz$ 都是不恰当的.

定理 24I(字母变换式的存在性) 设 φ 是公式，t 是项，x 是变量，那么可以找到一个公式 φ'(与 φ 的不同之处仅在于约束变量的选择) 使得：

(a) $\varphi \vdash \varphi'$ 且 $\varphi' \vdash \varphi$;

(b) 在 φ' 中用 t 可以替换 x.

126

证明 考虑给定的 t 与 x，在 φ 的基础上用递归构造 φ'. 对于原子公式 φ 这是很简单的，取 $\varphi'=\varphi$，那么 $(\neg\varphi)'=(\neg\varphi')$, $(\varphi\to\psi)'=(\varphi'\to\psi')$. 现在考虑 $(\forall y\varphi)'$ 的选择.

如果 y 不在 t 中出现，或者如果 $y=t$，那么可取 $(\forall y\varphi)'=(\forall y\varphi')$. 对于一般情形，就需要改变变量了.

选择不在 φ', t 或者 x 中出现的变量 z，定义 $(\forall y\varphi)'=\forall z(\varphi')^y_z$. 验证 (b)：注意到 z 不在 t 中出现，且在 φ' 中用项 t 可以替换 x(根据归纳假设). 因此，(因为 $x\neq z$) 在 $(\varphi')^y_z$ 中 t 可以替换 x. 为了验证 (a)，做如下计算：

$\varphi\vdash\varphi'$	归纳假设,
$\therefore \forall y\varphi\vdash\forall y\varphi'.$	
$\forall y\varphi'\vdash(\varphi')^y_z$	因为变量 z 不在 φ' 中出现,
$\therefore \forall y\varphi'\vdash\forall z(\varphi')^y_z$	由概化定理,
$\therefore \forall y\varphi\vdash\forall z(\varphi')^y_z.$	

同时,

$$\forall z(\varphi')^y_z \vdash ((\varphi')^y_z)^z_y, \qquad \text{由习题 9 可知这正是 } \varphi',$$
$$\varphi' \vdash \varphi; \qquad \text{归纳假设,}$$
$$\therefore \forall z(\varphi')^y_z \vdash \varphi$$
$$\therefore \forall z(\varphi')^y_z \vdash \forall y\varphi \qquad \text{由概化定理,}$$

最后一步使用了如下的事实：除非 $y = z$，那么 y 不会在 $(\varphi')^y_z$ 中自由出现，因此，也不在 $\forall z(\varphi')^y_z$ 中自由出现. ■

在上述定理证明中构造的公式 φ'，称为 φ 的 *字母变换式*. 该定理的含义是：在无法进行替换时，选择正确的字母变换式可以解决问题.

2.4.7 相等

这里给出一些下一节需要的关于相等的知识 (假定语言中包含 =). 首先，由 $v_1 = v_2$ 给定的关系是自反的，对称的和传递的 (也就是等价关系)：

$$\text{Eq1}: \quad \vdash \forall x x = x.$$

证明 第 5 组公理. ■

127

$$\text{Eq2}: \quad \vdash \forall x\forall y(x = y \to y = x).$$

证明 见例 (EQ2). ■

$$\text{Eq3}: \quad \vdash \forall x\forall y\forall z(x = y \to y = z \to x = z).$$

证明 习题 11. ■

另外，相等与谓词和函数符号是兼容的：

Eq4 (对二元谓词符号 P)：

$$\vdash \forall x_1\forall x_2\forall y_1\forall y_2(x_1 = y_1 \to x_2 = y_2 \to Px_1x_2 \to Py_1y_2).$$

对 n 元谓词符号也是类似的.

证明 需要证明

$$\{x_1 = y_1, x_2 = y_2, Px_1x_2\} \vdash Py_1y_2.$$

这可以通过对第 6 组公理的两个元素使用假言推理得到：

$$x_1 = y_1 \to Px_1x_2 \to Py_1x_2,$$
$$x_2 = y_2 \to Py_1x_2 \to Py_1y_2.$$
■

Eq5 (对二元函数符号 f)：

$$\vdash \forall x_1\forall x_2\forall y_1\forall y_2(x_1 = y_1 \to x_2 = y_2 \to fx_1x_2 = fy_1y_2).$$

对 n 元函数符号也是类似的.

证明 见例 (EQ5). ■

2.4.8 注记

通常一本逻辑书都会以演绎计算开始，首先介绍逻辑公理及其推理规则，并说明其可以为理智的人所接受．然后，继续证明一些可演绎计算的公式（或者从某些特定的非逻辑的公理可以演绎推导的公式，如集合论的公理．）

我们的观点有所不同，我们用另外一些事例来研究上一段描述的过程，并且只管运用正确的数学推理，无论这样的推理在所研究的推演运算中是否已经有相应的形式．

图 2-1 给出了两个层次的分界线：(a) 执行推理并证明结论的层次，(b) 我们关注的演绎计算的层次．

128

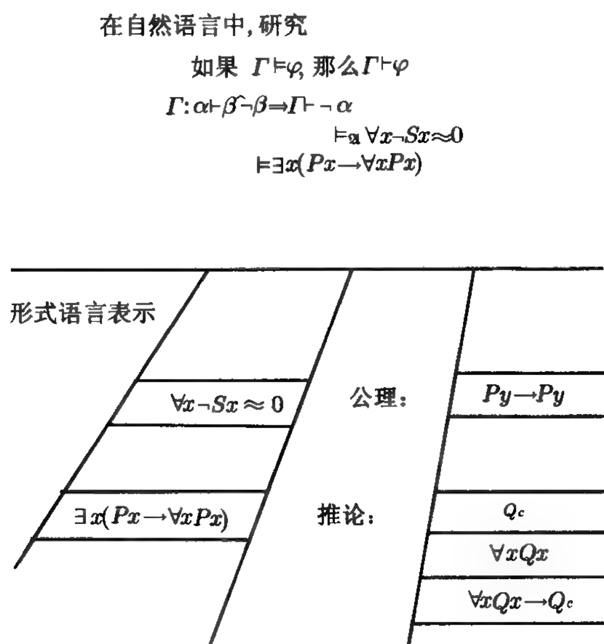


图 2-1 元语言（上），对象语言（下）

习题

1. 对项 u，令 u_t^x 是在 u 中将 x 替换为项 t 后得到表达式．请重述该定义（不使用任何"替换"或者与之同义的词语）．提示：对 u 使用归纳法．（注意：由新的定义可以清楚地看出 u_t^x 本身也是项．）
2. 下述公式各属于哪组公理（若可能）？
   (a) $[(\forall xPx \to \forall yPy) \to Pz] \to [\forall xPx \to (\forall yPy \to Pz)]$.
   (b) $\forall y[\forall x(Px \to Px) \to (Pc \to Pc)]$.
   (c) $\forall x\exists yPxy \to \exists yPyy$.
3. (a) 设 $\mathfrak{A}$ 是结构，且令 $s: V \to |\mathfrak{A}|$，在基本公式集上定义真值指派 v 如下：

$$v(\alpha) = T \quad \text{iff} \quad \models_{\mathfrak{A}} \alpha[s].$$

129

证明：对于任意公式（基本与否均可），有

$$\bar{v}(\alpha) = T \quad \text{iff} \quad \models_{\mathfrak{A}} \alpha[s].$$

说明：这个结论说明联结词 ¬ 与 → 在第 2 章与第 1 章中的含义是一致的．

(b) 证明：如果 Γ 重言蕴涵 φ，那么 Γ 逻辑蕴涵 φ．

4. 给出公式 $\forall x\varphi \to \exists x\varphi$ 的 (从 $\varnothing$) 一个演绎. (注意: 不只要证明演绎存在, 而且要写出整个演绎过程.)
5. 给出一个函数 f 使得: 如果一个公式 φ 有一个从集合 Γ 开始的长度为 n 的演绎, 且 x 不在 Γ 中自由出现, 那么 $\forall x\varphi$ 有从集合 Γ 开始的长度为 $f(n)$ 的公式. 给出的函数增长速度慢者为优.
6. (a) 证明如果 $\vdash \alpha \to \beta$, 那么 $\vdash \forall x\alpha \to \forall x\beta$.
   (b) 证明: 一般地, $\alpha \to \beta \vDash \forall x\alpha \to \forall x\beta$ 是不成立的.
7. (a) 证明: $\vdash \exists x(Px \to \forall xPx)$.
   (b) 证明: $\{Qx, \forall y(Qy \to \forall zPz)\} \vdash \forall xPx$.
8. (Q2b) 设 x 不在 α 中自由出现, 证明

$$\vdash (\alpha \to \exists x\beta) \leftrightarrow \exists x(\alpha \to \beta).$$

   在同样的假设下, 证明 Q3a:

$$\vdash (\forall x\beta \to \alpha) \leftrightarrow \exists x(\beta \to \alpha).$$

9. (再替换引理)(a) 试证 $(\varphi^x_y)^y_x$ 与 φ 一般不相等. 而且以下两种情况均有可能发生: x 在 $(\varphi^x_y)^y_x$ 中的某个位置上出现, 而在 φ 中同样的位置上不出现; 或者反过来, x 在 φ 中的某个位置上出现, 而在 $(\varphi^x_y)^y_x$ 中同样的位置上不出现.
   (b) 证明: 如果 y 不在 φ 中出现, 那么在 φ^x_y 和 $(\varphi^x_y)^y_x = \varphi$ 中 x 可替换 y. 提示: 对 φ 使用归纳法.
10. 证明:

$$\forall x\forall yPxy \vdash \forall y\forall xPyx.$$

11. (Eq3) 证明:

$$\vdash \forall x\forall y\forall z(x = y \to y = z \to x = z).$$

12. 证明: 任意和谐的公式集 Γ 可以扩充到和谐的公式集 Δ, 满足以下性质: 对于任意的公式 α, 或者 $\alpha \in \Delta$, 或者 $(\neg\, \alpha) \in \Delta$, 二者必居其一. (设语言是可数的. 不能使用命题逻辑的紧致性定理.)
13. 证明: $\vdash Py \leftrightarrow \forall x(x = y \to Px)$.

130

   说明: 更一般地, 如果在 φ 中 t 可以替换 x, 且 x 不在 t 中出现, 那么

$$\vdash [\varphi^x_t \leftrightarrow \forall x(x = t \to \varphi)].$$

   这样, 公式 $\forall x(x = t \to \varphi)$ 为替换 φ^x_t 提供了一个选择.
14. 证明: $\vdash (\forall x((\neg\, Px) \to Qx) \to \forall y((\neg\, Qy) \to Py))$.
15. 证明存在从 $\varnothing$ 到下列公式的演绎.
   (a) $\exists x\alpha \vee \exists x\beta \leftrightarrow \exists x(\alpha \vee \beta)$.
   (b) $\forall x\alpha \vee \forall x\beta \to \forall x(\alpha \vee \beta)$.
16. 证明存在从 $\varnothing$ 到下列公式的演绎.
   (a) $\exists x(\alpha \wedge \beta) \to \exists x\alpha \wedge \exists x\beta$.
   (b) $\forall x(\alpha \wedge \beta) \leftrightarrow \forall x\alpha \wedge \forall x\beta$.
17. 证明存在从 $\varnothing$ 到下列公式的演绎.
   (a) $\forall x(\alpha \to \beta) \to (\exists x\alpha \to \exists x\beta)$.
   (b) $\exists x(Py \wedge Qx) \leftrightarrow Py \wedge \exists xQx$.

2.5 可靠性与完备性理论

本节我们来学习两个主要的定理: 演绎计算的可靠性 ($\Gamma \vdash \varphi \Rightarrow \Gamma \vDash \varphi$) 及其完备性 ($\Gamma \vDash \varphi \Rightarrow \Gamma \vdash \varphi$). 然后学习一些有意义的结论, 包括紧致性和可枚举定理. 尽管演绎计算在

某种程度上是随机选择的，但值得注意的是某些演绎计算还是可靠的、完备的．这就能够激励勤奋的数学家们去关注源于公理的证明的存在性 (见 2.6.5 节).

可靠性定理 如果 $\Gamma \vdash \varphi$，那么 $\Gamma \vDash \varphi$.

可靠性定理说明演绎计算只能够得到“正确”的结论——演绎计算似乎没有意义．证明的基本思想是逻辑公理能够被逻辑蕴涵，且假言推理能够保持逻辑蕴涵．

引理 25A 逻辑公理都是恒真的．

完备性定理的证明 (基于上述引理) 使用归纳法，任何由 Γ 演绎得到的公式 φ 都是 Γ 逻辑蕴涵的．

情形 1：φ 就是逻辑公理．由引理，$\vDash \varphi$，因此 $\Gamma \vDash \varphi$.

情形 2：$\varphi \in \Gamma$，显然，$\Gamma \vDash \varphi$.

情形 3：φ 可以由 $\psi, \psi \to \varphi$ 通过假言推理得到，由归纳假设，这里 $\Gamma \vDash \psi$ 且 $\Gamma \vDash (\psi \to \varphi)$．于是，$\Gamma \vDash \varphi$. ■

131

接下来，我们需要证明引理．由 2.2 节的习题 6 可知，任何恒真公式的概化都是恒真的．因此我们只需考虑逻辑公理，而无需考虑其他的．这里，我们考察各组公理．

第 3 组公理：见 2.2 节习题 3.

第 4 组公理：见 2.2 节习题 4.

第 5 组公理：显而易见，$\mathfrak{A}$ 以 s 满足 $x = x$ 当且仅当 $s(x) = s(x)$，这自然是正确的．

第 1 组公理：由前一节的习题 3 可知，如果 $\varnothing$ 重言蕴涵 α，那么 $\varnothing \vDash \alpha$.

第 6 组公理(作为例子，见 2.2 节习题 5)：设 α 是原子的，且 α' 可以由 α 在某个地方通过将 x 替换为 y 得到．这足以证明

$$\{x = y, \alpha\} \vDash \alpha'.$$

因此，取任意的 $\mathfrak{A}$, s 使得

$$\vDash_{\mathfrak{A}} x = y[s], \quad 即\ s(x) = s(y).$$

那么，任何项 t 具有属性：如果 t' 可以由 t 通过将某些位置的 x 替换为 y 得到，那么 $\bar{s}(t) = \bar{s}(t')$．这是很明显的，完整的证明可以使用对 t 的归纳法．

如果 α 是 $t_1 = t_2$，那么 α' 肯定是 $t_1' = t_2'$，其中 t_i' 由 t_i 得到．

$$\begin{aligned} \vDash_{\mathfrak{A}} \alpha[s] \quad & \text{iff} \quad \bar{s}(t_1) = \bar{s}(t_2), \\ & \text{iff} \quad \bar{s}(t_1') = \bar{s}(t_2'), \\ & \text{iff} \quad \vDash_{\mathfrak{A}} \alpha'[s]. \end{aligned}$$

类似地，若 α 是 $Pt_1t_2\cdots t_n$，那么 α' 是 $Pt_1't_2'\cdots t_n'$，可以类似地讨论．

最后来看第 2 组公理．首先看一个简单的情形：证明 $\forall x Px \to Pt$ 是恒真的．设

$$\vDash_{\mathfrak{A}} \forall x Px[s].$$

那么对 $|\mathfrak{A}|$ 中任意的 d,

$$\models_{\mathfrak{A}} Px[s(x|d)].$$

特别地取 $d = \bar{s}(t)$:

$$\models_{\mathfrak{A}} Px[s(x|\bar{s}(t))]. \tag{a}$$

132 由原子公式的可满足性的定义，这等价于

$$\bar{s}(t) \in P^{\mathfrak{A}},$$

而这又等价于

$$\models_{\mathfrak{A}} Pt[s]. \tag{b}$$

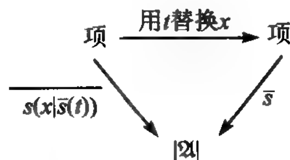
将这个过程应用到非原子公式的情形，我们需要从 (a) 到 (b) 的转换，这可以通过如下的替换引理得到：只要 t 对 φ 中的 x 是可替换的，那么

$$\models_{\mathfrak{A}} \varphi[s(x|\bar{s}(t))] \quad \text{iff} \quad \models_{\mathfrak{A}} \varphi^x_t[s]$$

考虑固定的 $\mathfrak{A}$ 和 s，对任何项 u，令 u^x_t 是在 u 中将 x 换为 t 后的结果.

引理 25B　$\bar{s}(u^x_t) = \overline{s(x|\bar{s}(t))}(u)$.

这个公式看上去比其含义要复杂的多，其含义是替换项 u 和 s 中的执行结果是等价的. 对应的交换图如下：



证明　对项 u 使用归纳法. 如果 u 是常数符号或者除 x 外的其他变量，那么 $u^x_t = u$，要求的等式就变成 $\bar{s}(u) = \bar{s}(u)$. 若 $u = x$，那么等式简化为 $\bar{s}(t) = \bar{s}(t)$. 尽管归纳步骤难以写出，但是在数学上却是显而易见的. ■

替换引理的本质是相似，其表明替换 φ 和 s 中的执行结果是等价的. 作为例子，读者可以参考 2.2 节的习题 10.

替换引理　如果项 t 可以替换合式公式 φ 中的变量 x，那么

$$\models_{\mathfrak{A}} \varphi^x_t[s] \quad \text{iff} \quad \models_{\mathfrak{A}} \varphi[s(x|\bar{s}(t))].$$

证明　使用对 φ 的归纳法证明上式对每个 s 成立.
情形 1：φ 是原子的. 那么由前面的引理即得. 例如，对某个项 u，如果 φ 是 Pu，那么

$$\begin{aligned}
\models_{\mathfrak{A}} Pu^x_t[s] \quad & \text{iff} \quad \bar{s}(u^x_t) \in P^{\mathfrak{A}}, \\
& \text{iff} \quad \overline{s(x|\bar{s}(t))}(u) \in P^{\mathfrak{A}} \quad \text{由引理 25B}, \\
& \text{iff} \quad \models_{\mathfrak{A}} Pu[s(x|\bar{s}(t))].
\end{aligned}$$

133

情形 2：φ 是 $\neg\psi$ 或者 $\psi\to\theta$. 那么对 φ 的结论可以由对 ψ 及 θ 的归纳假设得到.

情形 3：φ 是 $\forall y\psi$, 且 x 不在 φ 中自由出现. φ^x_t 就是 φ, 结论显然成立.

情形 4：φ 是 $\forall y\psi$, 且 x 在 φ 中自由出现. 由于在 φ 中 t 可以替换 x, y 不在 t 中出现, 且在 ψ 中 t 可以替换 x(见"可替换"的定义).
由第 1 条, 对 $|\mathfrak{A}|$ 中任意的 d, 有

$$\bar{s}(t)=\overline{s(y|d)}(t) \qquad (*)$$

由于 $x\neq y,\varphi^x_t=\forall y\psi^x_t$.

$$\begin{aligned}\models_{\mathfrak{A}}\varphi^x_t[s]\quad &\text{iff}\quad \text{对每一个 } d,\models_{\mathfrak{A}}\psi^x_t[s(y|d)],\\ &\text{iff}\quad \text{对每一个 } d,\models_{\mathfrak{A}}\psi[s(y|d)(x|\bar{s}(t))], \text{由演绎假设和 (*) 式},\\ &\text{iff}\quad \models_{\mathfrak{A}}\varphi[s(x|\bar{s}(t))].\end{aligned}$$

由归纳法知, 引理对任意的 φ 成立. ■

第 2 组公理：设在 φ 中 t 可以替换 x, 假定 $\mathfrak{A}$ 能够以 s 满足 $\forall x\varphi$. 我们需要证明 $\models_{\mathfrak{A}}\varphi^x_t[s]$. 对于 $|\mathfrak{A}|$ 中任意的 d,

$$\models_{\mathfrak{A}}\varphi[s(x|d)],$$

特别地, 令 $d=\bar{s}(t)$：

$$\models_{\mathfrak{A}}\varphi[s(x|\bar{s}(t))],$$

因此, 由替换引理,

$$\models_{\mathfrak{A}}\varphi^x_t[s].$$

这样, $\forall x\varphi\to\varphi^x_t$ 是恒真的.

这就证明了所有逻辑公理的证明是恒真的, 这也就证明了可靠性定理.

推论 25C 如果 $\vdash(\varphi\leftrightarrow\psi)$, 那么 φ 和 ψ 是逻辑等价的.

推论 25D 如果 φ' 是 φ 的字母变换式 (见定理 24I), 那么 φ 和 φ' 是逻辑等价的.

回忆一下, 集合 Γ 是和谐的当且仅当不存在这样的公式 φ 使得 $\Gamma\vdash\varphi$ 且 $\Gamma\vdash\neg\varphi$. 定义 Γ 是*可满足的*当且仅当存在某个 $\mathfrak{A}$ 和 s 使得 $\mathfrak{A}$ 以 s 满足 Γ 中的每个元素.

推论 25E 如果 Γ 是可满足的, 那么 Γ 是和谐的.

这个推论事实上等价于可靠性定理, 请读者自行验证. [134]
完备性定理是可靠性定理的逆命题, 也是一个更深入的结论.

完备性定理 (哥德尔, 1930)
(a) 如果 $\Gamma\models\varphi$, 那么 $\Gamma\vdash\varphi$;
(b) 任意和谐的公式集都是可满足的.

事实上, (a) 与 (b) 是等价的；参见习题 2. 因此只需证明 (b). 我们给出对于可数语言的证明, 稍后, 我们会指出对于更大基数的语言, 证明需要做改进. (可数语言是指只有可数多个符号, 或者等价地说, (根据定理 0B) 只有可数多个合式公式.)

证明的思想与命题逻辑的紧致性定理的证明相关. 我们从和谐集 Γ 开始, 在第 1~3 步中, 我们将其扩充到公式集 Δ, 使其满足

(i) $\Gamma \subseteq \Lambda$.

(ii) Δ 是和谐的, 也是最大的 (即对于任意的公式 α, 或者 $\alpha \in \Delta$ 或者 $(\neg\, \alpha) \in \Delta$, 二者必居其一).

(iii) 对于任意的公式 φ 和变量 x, 存在常数 c 使得

$$(\neg\, \forall x \varphi \to \neg\, \varphi_c^x) \in \Delta.$$

第 4 步构造结构 $\mathfrak{A}$, 使其满足 Γ 中不包含 "=". $|\mathfrak{A}|$ 是项的集合, 对于谓词符号 P,

$$\langle t_1, \cdots, t_n \rangle \in P^{\mathfrak{A}} \quad \text{iff} \quad Pt_1, \cdots, t_n \in \Delta.$$

最后, 第 5 步和第 6 步修改 $\mathfrak{A}$ 使其适合包含等号的公式的情形.

建议读者在第一次阅读时将大部分的步骤细节省去, 在了解了大概步骤后, 再来阅读整个证明. (证明的非细节部分, 我们会在左边使用竖线标出.)

证明　设 Γ 是可数语言的和谐公式集.

第 1 步: 通过加入可数的无限大的新常数符号集对语言进行扩充, 那么 Γ 在新的语言中保持和谐性.

细节: 如果不是这样, 那么对某个 β, 存在 (扩充语言中的) 从 Γ 到 $(\beta \wedge \neg\, \beta)$ 的演绎. 这个演绎包含了有限多个新的常数符号. 由定理 24F, 每个都可以使用变量替换. 因此, 可以得到一个在原来语言中的从 Γ 到 $(\beta' \wedge \neg\, \beta')$ 的演绎. 这与 Γ 是和谐公式集相矛盾.

135

第 2 步: 对每个 (新语言中的) 合式公式和每个变量 x, 我们在 Γ 中加入合式公式

$$\neg\, \forall x \varphi \to \neg\, \varphi_c^x,$$

其中 c 是一个新的常数符号. (基本思想是如果 φ 存在反例, 则用 c 命名.) Γ 与所有新加入的合式公式的集合 Θ 一起还是和谐的.

细节: 选取一个固定的 $\langle \varphi, x \rangle$ 可数序列, 这里 φ 是 (扩充语言的) 合式公式, x 是一个变量:

$$\langle \varphi_1, x_1 \rangle, \langle \varphi_2, x_2 \rangle, \langle \varphi_3, x_3 \rangle, \cdots.$$

这总是可以做到的, 因为语言是可数的. 设 θ_1 是

$$\neg\, \forall x_1 \varphi_1 \to \neg\, {\varphi_1}_{c_1}^{x_1},$$

这里的 c_1 是不在 φ_1 中出现的第一个新的常数符号. 那么继续对 $\langle \varphi_2, x_2 \rangle$ 定义 θ_2. 一般地, θ_n 可定义为

$$\neg\, \forall x_n \varphi_n \to \neg\, {\varphi_n}_{c_n}^{x_n},$$

其中 c_n 是不在 φ_n 中出现的第一个新的常数符号, 或者是对任意 $k < n$ 的 θ_k.

设 Θ 是集合 $\{\theta_1,\theta_2,\cdots\}$，那么 $\Gamma\cup\Theta$ 是和谐的. 否则，由于演绎是有限的，对于某个 $m\geqslant 0$，

$$\Gamma\cup\{\theta_1,\cdots,\theta_m,\theta_{m+1}\}$$

是不和谐的. 取满足条件的最小的 m，那么由 RAA

$$\Gamma\cup\{\theta_1,\cdots,\theta_m\}\vdash\neg\,\theta_{m+1}.$$

现在对某个 x, φ 及 c, θ_{m+1} 是

$$\neg\,\forall x\varphi\to\neg\,\varphi_c^x$$

由规则 T，可以得到两个事实：

$$\Gamma\cup\{\theta_1,\cdots,\theta_m\}\vdash\neg\,\forall x\varphi,$$
$$\Gamma\cup\{\theta_1,\cdots,\theta_m\}\vdash\varphi_c^x. \qquad (*)$$

因为 c 是不在任何左边的公式中出现的，那么可以应用推论 24G 到上述第二个式子中，得到

$$\Gamma\cup\{\theta_1,\cdots,\theta_m\}\vdash\forall x\varphi.$$

这和式 (*) 中的 m 是最小的矛盾 (或者若 $m=0$，则与 Γ 的和谐性矛盾). 136

第 3 步：扩充和谐集合 $\Gamma\cup\Theta$ 到最大和谐集 Δ，其中对于任意合式公式 φ，或者 $\varphi\in\Delta$ 或者 $(\neg\,\varphi)\in\Delta$，二者必居其一.

细节：可以模仿 1.7 节中的关于命题紧致性的证明方法来证明. 或者作如下讨论：设 Λ 是扩充语言的逻辑公理的集合. 由于 $\Gamma\cup\Theta$ 是和谐的，不存在公式 β 使得 $\Gamma\cup\Theta\cup\Lambda$ 同时重言蕴涵 β 和 $\neg\,\beta$. (这是根据定理 24B 而来，这里用到了命题逻辑的紧致性定理.) 因此，对所有基本公式的集合存在真值指派 v 满足 $\Gamma\cup\Theta\cup\Lambda$. 设

$$\Delta=\{\varphi|\bar{v}(\varphi)=T\}.$$

显然，对任意的 φ，或者 $\varphi\in\Delta$ 或者 $(\neg\,\varphi)\in\Delta$，二者必居其一，但二者不能够同时成立. 这样，我们有

$$\begin{aligned}\Delta\vdash\varphi\ &\Rightarrow\ \Delta\quad \text{重言蕴涵}\varphi\ (\text{由于}\Lambda\subseteq\Delta),\\ &\Rightarrow\ \bar{v}(\varphi)=T\qquad \text{由于 } v \text{ 满足}\Delta,\\ &\Rightarrow\ \varphi\in\Delta.\end{aligned}$$

因此，Δ 是和谐的，除非 φ 与 $\neg\,\varphi$ 同时属于 Δ.

事实上，不管 Δ 如何构造，它必定是演绎封闭的. 即

$$\begin{aligned}\Delta\vdash\varphi\ &\Rightarrow\ \Delta\nvdash\neg\,\varphi\qquad \text{由和谐性}\\ &\Rightarrow\ (\neg\,\varphi)\notin\Delta,\\ &\Rightarrow\ \varphi\in\Delta\qquad \text{由最大性}\end{aligned}$$

第 4 步：对新的语言，从 Δ 取结构 $\mathfrak{A}$，其中等号使用一个新的二元谓词符号 E 取代. $\mathfrak{A}$ 本身不是一个 Γ 能够满足的结构，但是一个初步的结构.

(a) $|\mathfrak{A}|$ = 新语言的所有项的集合.

(b) 定义二元关系 $E^{\mathfrak{A}}$：

$$\langle u,t\rangle \in E^{\mathfrak{A}} \text{ iff 公式 } u=t \text{ 属于 } \Delta.$$

(c) 对每个 n 元谓词参数 P，定义 n 元关系 $P^{\mathfrak{A}}$：

$$\langle t_1,\cdots,t_n\rangle \in P^{\mathfrak{A}} \text{ iff 公式 } Pt_1\cdots t_n \in \Delta.$$

137

(d) 对 n 元函数符号 f，设 $f^{\mathfrak{A}}$ 是如下定义的函数

$$f^{\mathfrak{A}}(t_1,\cdots,t_n) = ft_1\cdots t_n.$$

这包括了 $n=0$ 的情形：对于一个常数符号 c，取 $c^{\mathfrak{A}}=c$. 定义函数 $s:V\to|\mathfrak{A}|$，即 V 上的恒等函数 $s(x)=x$.

这样，对于任何项 t，$\bar{s}(t)=t$，对于任意合式公式 φ，设 φ^* 是将 φ 中用 E 替代等号后得到的结果. 那么

$$\models_{\mathfrak{A}} \varphi^*[s] \quad \text{iff} \quad \varphi\in\Delta.$$

细节：$\bar{s}(t)=t$ 可以使用对 t 的归纳法进行证明，但是整个证明都是直截了当的.

对于另外一个结论

$$\models_{\mathfrak{A}} \varphi^*[s] \quad \text{iff} \quad \varphi\in\Delta,$$

我们对出现在公式中的联结词或者量词的数目使用归纳法.

情形 1：原子公式. 我们可以以这样的方式定义 $\mathfrak{A}$，使得这种情况是显然的. 比如，如果 φ 是 Pt，那么

$$\begin{aligned}\models_{\mathfrak{A}} Pt[s] \quad &\text{iff} \quad \bar{s}(t)\in P^{\mathfrak{A}},\\ &\text{iff} \quad t\in P^{\mathfrak{A}},\\ &\text{iff} \quad Pt\in\Delta.\end{aligned}$$

类似地，

$$\begin{aligned}\models_{\mathfrak{A}} uEt[s] \quad &\text{iff} \quad \langle \bar{s}(u),\bar{s}(t)\rangle \in E^{\mathfrak{A}},\\ &\text{iff} \quad \langle u,t\rangle\in E^{\mathfrak{A}},\\ &\text{iff} \quad u=t\in\Delta.\end{aligned}$$

情形 2：否定.

$$\begin{aligned}\models_{\mathfrak{A}} (\neg\,\varphi)^*[s] \quad &\text{iff} \quad \not\models_{\mathfrak{A}} \varphi^*[s],\\ &\text{iff} \quad \varphi\notin\Delta \quad \text{由演绎假设},\\ &\text{iff} \quad (\neg\,\varphi)\in\Delta \quad \text{由}\Delta\text{的性质}.\end{aligned}$$

情形 3：条件.

$$
\begin{aligned}
\models_{\mathfrak{A}} (\varphi \to \psi)^*[s] \quad & \text{iff} \quad \not\models_{\mathfrak{A}} \varphi^*[s] \text{ 或者 } \models_{\mathfrak{A}} \psi^*[s], \\
& \text{iff} \quad \varphi \notin \Delta \text{ 或者 } \psi \in \Delta \quad \text{由演绎假设}, \\
& \text{iff} \quad (\neg \varphi) \in \Delta \text{ 或者} \psi \in \Delta, \\
& \Rightarrow \quad \Delta \vdash (\varphi \to \psi), \qquad \text{重言式}, \\
& \Rightarrow \quad \varphi \notin \Delta \text{ 或者 } [\varphi \in \Delta \text{ 且 } \Delta \vdash \psi], \\
& \Rightarrow \quad (\neg \varphi) \in \Delta \text{ 或者 } \psi \in \Delta,
\end{aligned}
$$

循环结束．且

$$\Delta \vdash (\varphi \to \psi) \quad \text{iff} \quad (\varphi \to \psi) \in \Delta.$$

(这可与 1.7 节的习题 2 进行比较．)

情形 4：量化．我们希望证明：

138

$$\models_{\mathfrak{A}} \forall x \varphi^*[s] \quad \text{iff} \quad \forall x \varphi \in \Delta.$$

(由于 $\forall x(\varphi^*)$ 与 $(\forall x\varphi)^*$ 是相同的，因此这里的不同符号的记法无关紧要．) Δ 包含合式公式 θ：

$$\neg \forall x \varphi \to \neg \varphi_c^x.$$

为了证明

$$\models_{\mathfrak{A}} \forall x \varphi^*[s] \Rightarrow \forall x \varphi \in \Delta,$$

我们的证明如下：如果 φ^* 总是正确的，那么对 c 也是正确的．因此由归纳假设可知 $\varphi_c^x \in \Delta$．但是 $\forall x\varphi \in \Delta$，因为 c 是作为 φ 的反例所选择的 (如果存在的话)．具体如下：

$$
\begin{aligned}
\models_{\mathfrak{A}} \forall x \varphi^*[s] \quad & \Rightarrow \quad \models_{\mathfrak{A}} \varphi^*[s(x|c)] & & \\
& \Rightarrow \quad \models_{\mathfrak{A}} (\varphi^*)_c^x[s] & & \text{由替换引理} \\
& \Rightarrow \quad \models_{\mathfrak{A}} (\varphi_c^x)^*[s], & & \text{同上} \\
& \Rightarrow \quad \varphi_c^x \in \Delta & & \text{归纳假设} \\
& \Rightarrow \quad (\neg \varphi_c^x) \notin \Delta & & \text{和谐性} \\
& \Rightarrow \quad (\neg \forall x \varphi) \notin \Delta & & \text{因为 } \theta \in \Delta \text{ 且 } \Delta \text{ 是演绎封闭的} \\
& \Rightarrow \quad \forall x \varphi \in \Delta. & &
\end{aligned}
$$

(我们仅用到了 Θ，需要知道，如果 $(\neg \forall x\varphi) \in \Delta$，那么对某个特殊的 c，有 $(\neg \varphi_c^x) \in \Delta$．)

考查其反面．具体论证如下：

$$
\begin{aligned}
\not\models_{\mathfrak{A}} \forall x \varphi^*[s] \quad & \Rightarrow \quad \not\models_{\mathfrak{A}} \varphi^*[s(x|t)] & & \text{对某个 } t \\
& \rightsquigarrow \quad \not\models_{\mathfrak{A}} (\varphi_t^x)^*[s] & & \text{由替换引理} \\
& \Rightarrow \quad \varphi_t^x \notin \Delta & & \text{归纳假设} \\
& \rightsquigarrow \quad \forall x \varphi \notin \Delta & & \Delta \text{ 是演绎封闭的}
\end{aligned}
$$

问题在于两个蕴含式都要求在 φ 中 t 是可以替换 x 的. 这可能不对，但是我们能够作如下改进：将 φ 换作字母变换式 ψ，其中 t 可替换 x. 那么就有

$\nvDash_{\mathfrak{A}} \forall x\varphi^*[s]$	$\Rightarrow$	$\nvDash_{\mathfrak{A}} \varphi^*[s(x\vert t)]$	对某个固定的 t
	$\Rightarrow$	$\nvDash_{\mathfrak{A}} \psi^*[s(x\vert t)]$	由字母变换式的语义等价性 (推理 25D)
	$\Rightarrow$	$\nvDash_{\mathfrak{A}} (\psi^x_t)^*[s]$	由替换引理
	$\Rightarrow$	$\psi^x_t \notin \Delta$	由归纳假设
	$\Rightarrow$	$\forall x\psi \notin \Delta$	由于 Δ 的演绎封闭
	$\Rightarrow$	$\forall x\varphi \notin \Delta$	由字母变换式的语法等价性 (推理 24I)

这就完成了所有可能的情形；由归纳法，对于任意的 φ，

139

$$\vDash_{\mathfrak{A}} \varphi^*[s] \quad \text{iff} \quad \varphi \in \Delta.$$

如果我们最初的语言不包含等号，那么到这里就结束了. 因为我们只需要将结构 $\mathfrak{A}$ 限制到最初的语言，以得到一个满足 Γ 的每个元素的结构，该结构带有恒等函数.

然而，目前的假设是最初的语言也包含等号. 那么这样得到的结构 $\mathfrak{A}$ 就不够用了. 例如，如果 Γ 包含句子 $c = d$ (其中 c 和 d 是不同的常数符号)，那么我们就需要一个结构 $\mathfrak{B}$ 使得 $c^{\mathfrak{B}} = d^{\mathfrak{B}}$，得到的 $\mathfrak{B}$ 是 $\mathfrak{A}$ 模 $E^{\mathfrak{A}}$ 的商结构 $\mathfrak{A}/E$.

第 5 步：$E^{\mathfrak{A}}$ 是一个 $|\mathfrak{A}|$ 上的等价关系. 对其中每个 t，设 $[t]$ 是等价类. 事实上，$E^{\mathfrak{A}}$ 是 $\mathfrak{A}$ 的 同余关系. 这就意味着满足下述条件：

(i) $E^{\mathfrak{A}}$ 是 $|\mathfrak{A}|$ 上的等价关系.

(ii) 对于每个谓词符号 P，$P^{\mathfrak{A}}$ 与 $E^{\mathfrak{A}}$ 是相容的：

$$\langle t_1, \cdots, t_n\rangle \in P^{\mathfrak{A}} \text{ 且} t_i E^{\mathfrak{A}} t_i' ,\ 1 \leqslant i \leqslant n \Rightarrow \langle t_1', \cdots, t_n'\rangle \in P^{\mathfrak{A}}.$$

(iii) 对于每个函数符号 f，$f^{\mathfrak{A}}$ 与 $E^{\mathfrak{A}}$ 是相容的：

$$t_i E^{\mathfrak{A}} t_i' ,1 \leqslant i \leqslant n \Rightarrow f^{\mathfrak{A}}(t_1, \cdots, t_n) E^{\mathfrak{A}} f^{\mathfrak{A}}(t_1', \cdots, t_n').$$

在这些条件下，我们可以构造商结构 $\mathfrak{A}/E$，具体如下：

(a) $|\mathfrak{A}/E|$ 是所有 $|\mathfrak{A}|$ 的等价类的集合；

(b) 对每个 n 元谓词符号 P，

$$\langle [t_1], \cdots, [t_n]\rangle \in P^{\mathfrak{A}/E} \quad \text{iff} \quad \langle t_1, \cdots, t_n\rangle \in P^{\mathfrak{A}}.$$

(c) 对每个 n 元函数符号 f，

$$f^{\mathfrak{A}/E}([t_1], \cdots, [t_n]) = [f^{\mathfrak{A}}(t_1, \cdots, t_n)].$$

这包含了 $n = 0$ 的情形：

$$c^{\mathfrak{A}/E} = [c^{\mathfrak{A}}].$$

设 $h:|\mathfrak{A}|\to|\mathfrak{A}/E|$ 是自然映射：

$$h(t)=[t].$$

那么 h 是一个从 $\mathfrak{A}$ 到 $\mathfrak{A}/E$ 上的同态. 进一步，$E^{\mathfrak{A}/E}$ 是 $|\mathfrak{A}/E|$ 上的相等关系. 因此对所有 φ：

$$\begin{aligned}\varphi\in\Delta &\Leftrightarrow \models_{\mathfrak{A}}\varphi^*[s]\\ &\Leftrightarrow \models_{\mathfrak{A}/E}\varphi^*[h\circ s]\\ &\Leftrightarrow \models_{\mathfrak{A}/E}\varphi[h\circ s]\end{aligned}$$

因此 $\mathfrak{A}/E$ 以 $h\circ s$ 满足 Δ 的每个元素 (因此，也满足 Γ 的每个元素). 140

细节：回忆

$$\begin{aligned}tE^{\mathfrak{A}}t' \quad &\text{iff}\quad (t=t')\in\Delta,\\ &\text{iff}\quad \Delta\vdash t=t'.\end{aligned}$$

(i) 根据相等的性质 Eq1，Eq2 和 Eq3，$E^{\mathfrak{A}}$ 是 $\mathfrak{A}$ 上的一个等价关系；

(ii) 根据相等的性质 Eq4，$P^{\mathfrak{A}}$ 与 $E^{\mathfrak{A}}$ 是相容的；

(iii) 根据相等的性质 Eq5，$f^{\mathfrak{A}}$ 与 $E^{\mathfrak{A}}$ 是相容的.

由 $P^{\mathfrak{A}}$ 与 $E^{\mathfrak{A}}$ 是相容的，可得 $P^{\mathfrak{A}/E}$ 是良定义的. 类似地，由 $f^{\mathfrak{A}}$ 与 $E^{\mathfrak{A}}$ 是相容的，可得 $f^{\mathfrak{A}/E}$ 是良定义的.

从这个构造可以得到，h 是一个从 $\mathfrak{A}$ 到 $\mathfrak{A}/E$ 上的同态，且

$$\begin{aligned}[t]E^{\mathfrak{A}/E}[t'] \quad &\text{iff}\quad tE^{\mathfrak{A}}t',\\ &\text{iff}\quad [t]=[t'].\end{aligned}$$

最后，

$$\begin{aligned}\varphi\in\Delta &\Leftrightarrow \models_{\mathfrak{A}}\varphi^*[s] && \text{由第 4 步}\\ &\Leftrightarrow \models_{\mathfrak{A}/E}\varphi^*[h\circ s] && \text{由同态定理}\\ &\Leftrightarrow \models_{\mathfrak{A}/E}\varphi[h\circ s],\end{aligned}$$

最后一步可以根据 $E^{\mathfrak{A}/E}$ 是 $|\mathfrak{A}/E|$ 上的相等关系进行判断.

第 6 步：将结构 $\mathfrak{A}/E$ 限制到最初的语言上. 这个 $\mathfrak{A}/E$ 的限制以 $h\circ s$ 满足 Γ 的每个元素. ■

对不可数语言，需要对上述的完备性定理的证明做一些修改. 设语言的基数为 λ. (由此，该语言有 λ 符号或者 λ 公式.) 假定读者具有坚实的集合论知识，我们需要的修改如下. 在第 1 步，加入 λ 个新的常数符号，其余细节不需要改变. 在第 2 步，只需要改变细节. 基数 λ 是初始序数. (这里需要对语言做巧妙的排序.) “枚举” 序对

$$\langle\varphi_\alpha,x_\alpha\rangle_{\alpha<\lambda}$$

按照小于 λ 的序数排列. 对 $\alpha < \lambda$, θ_α 是

$$\neg \forall x_\alpha \varphi_\alpha \to (\neg \varphi)^{x_\alpha}_{c_\alpha},$$

其中 c_α 是第一个不在 φ_α 或者是对任意 $\beta < \alpha, \theta_\beta$ 中的新常数符号. (这最多排除了 $\aleph_0 \cdot \mathrm{card}(\alpha)$ 个常数符号, 因此还会剩下一些.) 最后, 第 3 步我们通过 Zorn 引理得到一个极大集合 Δ. 证明的其余部分就无需改动了.

141

紧致性定理　(a) 如果 $\Gamma \vDash \varphi$, 那么存在某个有限的 $\Gamma_0 \subseteq \Gamma$, 有 $\Gamma_0 \vDash \varphi$.
(b) 如果 Γ 的每个有限子集 Γ_0 都是可满足的, 那么 Γ 是可满足的.
特别地, 句子集 Σ 有模型当且仅当其每个有限子集有模型.

证明　为了证明 (a), 只需考察

$$\begin{aligned} \Gamma \vDash \varphi \quad &\Rightarrow \quad \Gamma \vdash \varphi \\ &\Rightarrow \quad \Gamma_0 \vdash \varphi \qquad \text{对某个有限的 } \Gamma_0 \subseteq \Gamma\text{, 演绎是有限的} \\ &\Rightarrow \quad \Gamma_0 \vDash \varphi. \end{aligned}$$

(b) 可以类似证明. 如果 Γ 的每个有限子集是可满足的, 那么由可靠性, 每个子集都是和谐的. 这样 Γ 就是和谐的, 因为演绎是有限的. 因此由完备性可知, Γ 是可满足的. (事实上 (a) 与 (b) 是等价的, 参见 1.7 节的习题 3.) ■

初次接触紧致性定理, 自然会试图合并 (根据某些代数或者集合论运算) 那些在各种有限子集中都能被满足的结构, 因此得到一个能够满足整个集合的结构. 实际上, 这样的证明是可能的; 所需使用的运算是*超积*. 但是, 这里我们不进一步讨论这个有趣的问题.

注意到紧致性定理仅涉及 2.2 节中的语义概念, 根本没有涉及演绎. 而存在能够避免演绎的证明. 相同的评论可以用到如下定理中去.

***可枚举定理**　对合理的语言, 恒真 (valid) 合式公式的集合是能行可枚举的.

合理的语言是指其参数集合能够能行枚举, 并满足如下两个关系:

$$\{\langle P, n\rangle | P \text{ 是 } n \text{ 元谓词符号 }\}$$

以及

$$\{\langle f, n\rangle | f \text{ 是 } n \text{ 元函数符号 }\}$$

是可判定的. 例如, 只有有限多个参数的任何语言 (这样的语言称作*有限语言*) 肯定是合理的, 因为有限集合总是可判定的. 另一方面, 合理的语言必定是可数的, 因为我们不能够能行枚举不可数集合. (事实上, 可以使用更强的表达: 如 1.7 节, 需要一个合适的输入/输出格式, 其使用的符号集是*有限的*, 其蕴涵了所有字符串集合的可数性.)

142

该定理的精确版本在 3.4 节中给出 (见其中的第 20 条). 两个不同版本的证明的实质是一样的.

证明　其本质是 Λ 是可判定的, 因而演绎的集合也是可判定的.

设给定表达式 ε. (已经有了合理性的假设，可给予别人的只有可数多个符合条件的对象.) 我们希望判定 ε 是否在 Λ 中. 首先，我们检查其作为公式的语法形式. (对于命题逻辑，这样的检查有详细的说明；见 1.3 节. 在 2.3 节给出了针对一阶逻辑的类似说明.) 如果 ε 通过检查，还要检查其是否是重言式的概化 (使用真值表). 如果不是，则检查 ε 是否具有第 2 组公理的语法形式，如此等等. 当我们检查完第 6 组公理时，如果 ε 不能够被接受，那么 ε 不在 Λ 中.

(上述内容是要判定读者能否区分 Λ 的元素. 读者如果还不能够判定，可返回去阅读 3.4 节的内容.)

由于 Λ 是可判定的，Λ 的恒真结论集是能行可枚举的，见定理 17G. 但是

$$\begin{aligned}&\{\alpha|\ \alpha\text{ 是}\Lambda\text{的一个恒真结论}\}\\&=\{\alpha|\vdash\alpha\}\text{由定理24B}\\&=\{\alpha|\alpha\text{恒真}\}\end{aligned}$$

■

这个证明的最后一段可以使用如下的论证取代，这样看起来或许更清楚明了. 首先，我们说 (从 $\varnothing$ 开始的) 演绎的集合是可以判定的. 对于给定的有限序列 $\alpha_0,\cdots,\alpha_n$，顺序检查每个 α_i，以判定其是否在 Λ 中或者可以通过序列中前面的元素经过假言推理得到. 然后枚举恒真式，从合式公式的能行序列开始. 检查每个序列，判定其是否是演绎. 如果不是，则丢弃. 如果是，将其最后一个元素置入恒真列表中. 连续使用这种方法 (尽管效率不高)，可以产生一个列表，其中任意恒真公式都会出现在此表中.

***推论 25F** 设 Γ 是合理语言的合式公式的可判定集.

143

(a) Γ 的定理集是能行可枚举的；

(b) 由 Γ 逻辑蕴涵的公式集 $\{\varphi|\Gamma\vDash\varphi\}$ 是能行可枚举的.

(当然，(a) 与 (b) 是指同样的集合，这个推论本身包含了可枚举定理，其中 $\Gamma=\varnothing$.)

证明 1 枚举恒真公式；只要发现如下形式

$$\alpha_n\to\cdots\to\alpha_1\to\alpha_0,$$

检查 $\alpha_n,\cdots,\alpha_1$ 是否在 Γ 中. 如果在，那么将 α_0 置入 Γ 的定理列表中. 用这种方法，Γ 的任何定理最终都能被列出. ■

证明 2 $\Gamma\cup\Lambda$ 是可判定的，因此其恒真的结果集也是能行可枚举的，且正是我们所需要的. ■

例如，设 Γ 是通常的集合论系统的 (可判定) 公理集合，那么这个推论告诉我们集合论的定理集都是能行可枚举的.

更一般地，在判定某些公理理论时，坚持公理集合是可判定的这一点是很自然的. 毕竟，我们希望由这些公理所得到的证明是可验证的且确定无疑的论证. 验证的部分过程涉及对公理的验证. 由于这个验证是可行的，所以公理集应该是可判定的 (或者至少是半可判定的). 这样可以得到一个结论：由公理集得到的定理集是能行可枚举的.

***推论 25G** 设 Γ 是合理语言中的可判定的公式集，对任意句子 σ，或者 $\Gamma\vDash\sigma$，或者 $\Gamma\vDash\neg\sigma$. 那么 Γ 蕴涵的句子集是可判定的.

证明　如果 Γ 是不和谐的，那么存在所有句子组成的可判定的集合．因此，可假定 σ 是和谐的，设给定 σ，要求判定 $\Gamma \vDash \sigma$ 是否成立．我们能够枚举 Γ 的定理，并检查 σ 或者 $\neg\sigma$ 是否在其中．最终，其中一个会出现在 Γ 中，即可得到结论．■

(注意到这个证明实际上描述了两个判定的过程．当 Γ 和谐时，一个是正确的；当 Γ 不和谐时，另一个是正确的．因此，任何一种情况，决策过程都存在．但是对于给定的 Γ 的有限描述，我们不能够有效地做出决定．一个集合是可判定的当且仅当对这个集合存在一个决策过程，但这并不意味着我们已经得到了这个决策．)

144

需要说明的是，一般地，我们对可枚举的证明不能够扩充到可判定性的证明．对于大多数的语言，恒真公式的集合是不可判定的．(见 3.5 节的丘奇定理．)

历史注记

(可数语言的) 完备性定理最早出现在 1930 年哥德尔的博士论文中．(与 1931 年发表的"哥德尔不完备定理"不会产生混淆，在第 3 章会考虑这个结果．)(可数语言的) 完备性定理是作为推论给出的．

1936 年马尔采夫的一篇论文中隐含了不可数语言的紧致性定理．其证明使用了 Skolem 函数 (见 4.2 节) 和命题逻辑的紧致性定理．1941 年，马尔采夫的一篇论文第一次将不可数语言的紧致性定理清楚地表述出来．

可枚举定理隐含在 1928 年斯科伦发表的结果中，在此基础上才有了 1930 年哥德尔的著作．

我们给出的完备性定理的证明最早出现在 1949 年 Leon Henkin 的毕业论文中，与哥德尔最初的证明不同，Henkin 很容易地将证明推广到了具有任意基数的语言．

习题

1. (语义规则 EI) 假设常数符号 c 不在 φ, ψ 和 Γ 中出现，$\Gamma; \varphi^x_c \vDash \psi$，证明：$\Gamma; \exists x\varphi \vDash \psi$．(不使用可靠性定理和完备性定理．)
2. 证明完备性定理 (a) 与 (b) 等价．提示：$\Gamma \vDash \varphi$ iff $\Gamma \cup \{\neg\varphi\}$ 是不可满足的，且 Δ 是可满足的当且仅当 $\Delta \nvDash \perp$，其中 $\perp$ 是某个不可满足的、可批驳的公式，如 $\neg\forall x x = x$．
   说明：类似地，可靠性定理等价于每个可满足的公式集是和谐的．
3. 设 $\Gamma \vdash \varphi$，且 P 是不在 Γ 和 φ 中出现的谓词符号．是否存在不出现 P 的从 Γ 到 φ 的演绎？提示：这个问题有两种不同的方法．"软"方法使用两个不同的语言，一个含有 P，另一个不含 P；"硬"方法则考虑是否能够可以从给定的从 Γ 到 φ 的演绎中消去 P．

145

4. 设 $\Gamma = \{\neg\forall v_1 Pv_1, Pv_2, Pv_3, \cdots\}$，$\Gamma$ 是否和谐？是否可满足？
5. 证明无限图可以使用四色标注当且仅当每个有限子图都可使用四色标注．提示：构造一个语言，其中为每个国家构造一个常数符号，每种颜色构造一个一元谓词符号，然后使用紧致性定理．
6. 设 Σ_1、Σ_2 是句子集，且二者没有共同的模型．证明存在一个句子 τ 使得 $\text{Mod}\ \Sigma_1 \subseteq \text{Mod}\ \tau$ 且 $\text{Mod}\ \Sigma_2 \subseteq \text{Mod}\ \neg\tau$．(也可表述为：不相交的 EC_Δ 类可以使用 EC 类进行区分．) 提示：$\Sigma_1 \cup \Sigma_2$ 是不可满足的，使用紧致性定理．
7. 完备性定理说明每个句子或者存在 (从 $\varnothing$) 演绎，或者存在否定模型 (即一个结构，该句子在其中是假的)．对每个下述句子，证明其演绎存在或者给出一个否定模型．
   (a) $\forall x(Qx \to \forall y Qy)$．
   (b) $(\exists x Px \to \forall y Qy) \to \forall z(Pz \to Qz)$．

(c) $\forall z(Pz \to Qz) \to (\exists x Px \to \forall y Qy)$.

(d) $\neg \exists y \forall x(Pxy \leftrightarrow \neg Pxx)$.

8. 设 (带有等号的) 语言只有 $\forall$ 及 P 两个参数, 其中 P 是一个二元谓词符号. 设 $\mathfrak{A}$ 是结构且 $|\mathfrak{A}| = \mathbb{Z}$, 且 $\langle a, b\rangle \in P^{\mathfrak{A}}$ 当且仅当 $|a-b| = 1$. 这样 $\mathfrak{A}$ 就像一个无限图:



证明: 存在一个初等等价的不连通的结构 $\mathfrak{B}$. (连通的意思是指对 $|\mathfrak{A}|$ 的任意两个元素之间存在一条路. 一条从 a 到 b 的长度为 n 的路是一个序列 $\langle p_0, p_1, \cdots, p_n\rangle$, 其中 $a=p_0$, $b=p_n$ 且对每个 i, $\langle p_i, p_{i+1}\rangle \in P^{\mathfrak{A}}$.) 提示: 加入常数符号 c 和 d, 给出 c 与 d 分离的句子. 使用紧致性定理.

9. 在 2.4 节用到了一个特定的逻辑公理的集合 Λ, 对这个集合可以作有限制的变换.

(a) 如果在集合 Λ 中加入某个无效公式 ψ, 证明可靠性定理在这里失效.

(b) 另外一个极端, 如果取 $\Lambda = \varnothing$, 即不使用任何公理, 证明此时完备性定理失效.

(c) 在 Λ 中加入一个新的恒真公式, 可靠性定理和完备性定理都还成立, 请解释其原因. 146

2.6 理论的模型

本节我们不再讨论演绎和逻辑公理, 再返回来考虑 2.2 节讨论的问题. 当然, 有了上一节的定理, 我们就可以解决更多的问题了.

2.6.1 有限模型

有些句子只有无限模型, 譬如, $<$ 是没有最大元的序关系. 这样的句子的否定是有限**恒真的**, 即在每个有限结构中是真的.

也有的句子只有有限模型, 例如, 模型 $\forall x \forall y x = y$ 的基数就是 1. 但是如果所有的模型都是有限的, 那么根据如下定理模型的大小存在一个有限的界.

定理 26A 如果句子集合 Σ 有任意大的基数的有限模型, 那么就有一个无限模型.

证明 对每个整数 $k \geqslant 2$, 可以找到一个句子 λ_k "至少存在 k 个对象." 例如,

$$\lambda_2 = \exists v_1 \exists v_2 v_1 \neq v_2,$$
$$\lambda_3 = \exists v_1 \exists v_2 \exists v_3 (v_1 \neq v_2 \wedge v_1 \neq v_3 \wedge v_2 \neq v_3).$$

考虑集合

$$\Sigma \cup \{\lambda_2, \lambda_3, \cdots\}.$$

根据假设, 任意有限子集都有模型. 再由紧致性定理, 整个集合有一个模型, 显然这个模型是无限的. ■

比如, 在群论中, 存在某些非常巧妙的等式在每个有限群中都是真的, 但在每个无限群中都是假的, 这一点先前我们是深信不疑的. 但是根据上述定理, 这样的等式是不存在的.

定理的证明给出了一个用以求给定特性的结构的方法. 首先给出一些说明结构特性的句子 (可能是在扩充语言中的), 如果能够证明这些句子的任何有限子集都共有一个模型, 紧致性定理说明这个结构就是给定句子的模型. 接下来, 我们会看到一些使用这种方法的例子.

推论 26B (对于给定的语言) 所有有限结构的类不是 EC_Δ，所有无限结构的类不是

147

EC.

证明 前半部分可以直接从定理得到. 如果所有无限结构的类是 $\mathrm{Mod}\tau$，那么所有有限结构的类为 $\mathrm{Mod}\neg\tau$. 但是这个类连 EC_Δ 都不是，当然也不是 EC. ■

无限结构的类是 EC_Δ，即 $\mathrm{Mod}\ \{\lambda_2, \lambda_3, \cdots\}$.

接下来，我们考虑与有限结构相关的决策问题. 对于任意的结构 $\mathfrak{A}$，$\mathfrak{A}$ 的理论定义为 $\mathfrak{A}$ 中为真的所有句子的集合，记作：$\mathrm{Th}\ \mathfrak{A}$. 对于有限结构 $\mathfrak{A}$，$\mathrm{Th}\ \mathfrak{A}$ 是否可判定？句子集是否存在有限的可判定模型？

下面的考察可能会有助于对这些问题的回答.

(1) 任意有限的结构 $\mathfrak{A}$ 同构于论域为 $\{1, 2, \cdots, n\}$ 的结构，其中 n 为 $\mathfrak{A}$ 的大小 (即 $n = \mathrm{card}|\mathfrak{A}|$)，若 $|\mathfrak{A}| = \{a_1, \cdots, a_n\}$ ，那么这里仅是简单地把 a_i 替换为 i.

例如，设语言中仅有参数 $\forall$ 和一个二元谓词符号 E(对有向图中的边关系而言)，考虑有限结构 $\mathfrak{B}$，其论域 $|\mathfrak{B}|$ 包含 4 个不同的对象 $\{a, b, c, d\}$，且

$$E^{\mathfrak{B}} = \{\langle a, b\rangle, \langle b, a\rangle, \langle b, c\rangle, \langle c, c\rangle\}.$$

那么，$\mathfrak{B}$ 同构于结构

$$(\{1, 2, 3, 4, \}; \{\langle 1, 2\rangle, \langle 2, 1\rangle, \langle 2, 3\rangle, \langle 3, 3\rangle\}).$$

但是这里还有另外一种情况，如果我们将 $|\mathfrak{B}|$ 中的元素排列为 b, a, d, c，那么我们得到的同构结构就变成了

$$(\{1, 2, 3, 4, \}; \{\langle 1, 2\rangle, \langle 2, 1\rangle, \langle 1, 4\rangle, \langle 4, 4\rangle\}).$$

(2) 对于有限语言而言，有限结构可以使用一个有限的符号串表达. 如上例可以使用下面的一行符号串来表达：

$$(\{1, 2, 3, 4, \}; \{\langle 1, 2\rangle, \langle 2, 1\rangle, \langle 2, 3\rangle, \langle 3, 3\rangle\}),$$

当然，也可以使用基为 10(或者读者喜欢的其他基) 的数字、标点和分隔符 (如括号) 书写出来. 因此，这样的结构可以同他人或者机器进行交流. 有限符号串可以使用一种合适的输入格式进行书写.

(3) 对于有限语言，给定有限结构 $\mathfrak{A}$，论域为 $\{1, 2, \cdots, n\}$(由前面的考察，给出这样一个对象是可能的)，一个合式公式 φ，以及一个指派 s_φ 将论域中的数字指派给 φ 中的自由变量

148

(当然是有限多的)，那么就能够判定是否有 $\models_{\mathfrak{A}} \varphi[s_\varphi]$.

例如，给定

$$\mathfrak{B} = (\{1, 2, 3, 4, \}; \{\langle 1, 2\rangle, \langle 2, 1\rangle, \langle 2, 3\rangle, \langle 3, 3\rangle\}),$$

$$\varphi = \forall v_1((\neg \forall v_2 \neg Ev_2v_1) \to Ev_1v_1),$$

可以按照图 2-2 所示的树进行计算，我们发现句子 φ：“E 的值域中的任何对象都与其自身有关” 在 $\mathfrak{B}$ 中是错误的.

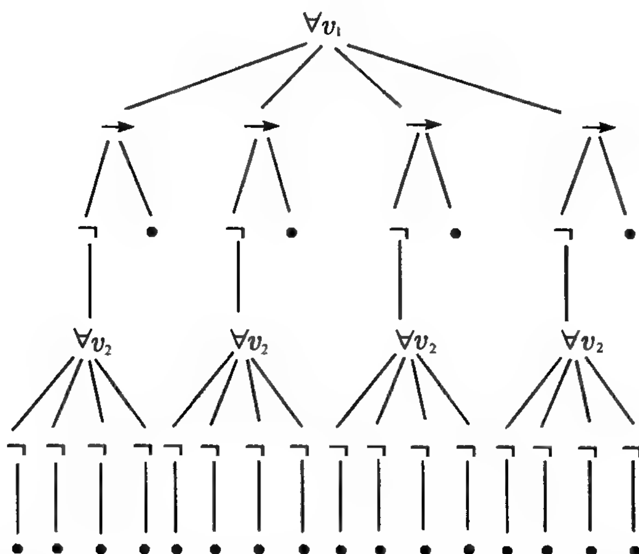


图 2-2 在论域大小为 4 的结构中检查句子 $\forall v_1((\neg\forall v_2 \neg Ev_2v_1) \to Ev_1v_1)$

在树中的每个叶子上 (即每个最小的节点) 有一个原子公式，检查其是否能够被满足. 要注意每个量词的出现都会导致对 n 个元素的论域进行一次检查. 对一个具有 k 个量词的公式 φ，树中的叶子数目的上界是 n 的 k 次方的多项式. 如果语言包含函数符号，那么其中的每个项都要使用结构给出的 (有限) 函数进行检查.

特别地，如果限制到句子上，对如上给定的 $\mathfrak{A}$ 和句子 σ，我们可以判定 $\mathfrak{A}$ 是否是 σ 的模型. (事实上，这里的 σ 甚至可以是第 4 章中要讨论的二阶句子.)

***定理 26C** 对有限语言中的有限结构 $\mathfrak{A}$，Th $\mathfrak{A}$ 是可判定的.

149

证明 1 由考察 1 可知，我们能够将 $\mathfrak{A}$ 替换成论域为 $\{1,\cdots,n\}$ 的同构结构，且在替换后，不改变句子的真值. 然后应用考察 3 即可. ■

证明 2 由 2.2 节的习题 17(a)，存在句子 $\delta_{\mathfrak{A}}$，它给出 $\mathfrak{A}$ 的一个同构. 这就得到

$$\text{Th}\,\mathfrak{A} = \{\sigma | \delta_{\mathfrak{A}} \vDash \sigma\}.$$

(细节：注意到对 “$\subseteq$”，如果 σ 在 $\mathfrak{A}$ 中是真的，那么其在 $\mathfrak{A}$ 的所有同构结构中都是真的，因此在 $\delta_{\mathfrak{A}}$ 的所有模型中也都是真的，因此 $\delta_{\mathfrak{A}} \vDash \sigma$. 反过来更简单，如果 $\delta_{\mathfrak{A}} \vDash \sigma$，那么 σ 在 $\delta_{\mathfrak{A}}$ 的所有模型中也都是真的，当然在 $\mathfrak{A}$ 中也是这样的.) 应用推论 25G，注意到对每个 σ，或者 $\vDash_{\mathfrak{A}} \sigma$ 或者 $\vDash_{\mathfrak{A}} \neg\sigma$. ■

(4) 给定句子 σ 和正整数 n，能够判定 σ 是否存在一个 n 元模型，即，二元关系

$$\{\langle\sigma,n\rangle | \sigma \text{ 有一个大小为 } n \text{ 的模型}\}$$

是可判定的.

这里的关键思想是只有有限多个结构需要检查，且我们能够进行检查. 由考察 1，句子 σ 具有大小为 n 的模型当且仅当其具有论域为 $\{1,\cdots,n\}$ 的模型. 如果将语言限制到仅出现在 σ 中的参数上，那么就只有有限多个这样的结构，就可以按照一定的方法系统生成. (例

如，如果参数只有 $\forall$ 和二元谓词符号，那么就有 2^{n^2} 个不同构的结构.）应用考察 3，就可以检查其中的任何一个是否是 σ 的模型.

(5) 句子 σ 的 *谱系*(spectrum) 定义为 $\{n|\sigma$ 具有大小为 n 的模型 $\}$，见 2.2 节的习题 16. 由考察 4 可以看出，任何句子的谱系都是可判定的正整数集合.

***定理 26D**　对于有限语言，$\{\sigma|\sigma$ 具有有限模型 $\}$ 是能行可枚举的.

证明　这里给出一个半判定的方法：给定 σ，由考察 4，首先检查其是否具有大小为 1 的模型. 如果没有，检查其是否具有大小为 2 的模型，如此持续下去. ■

***推论 26E**　设语言是有限的，且 Φ 是在每个有限结构中都是真的句子集合，那么其补 $\bar{\Phi}$ 是能行可枚举的.

证明　对于句子 σ，$\sigma \in \bar{\Phi} \Leftrightarrow (\neg\ \sigma)$ 具有有限模型，可以对 $(\neg\ \sigma)$ 使用上述半判定的判断方法. ■

150

由定理 17F，Φ 是可判定的当且仅当它是能行可枚举的，可惜情况不是这样. 对如下定理我们不加以证明.

***Trakhtenbrot 定理 (1950)**　句子集合

$$\Phi = \{\sigma|\sigma \text{ 在每个有限结构中是真的 }\}$$

通常是不可判定的，也不是能行可枚举的.

这样，与有限结构的可枚举定理类似的可判定的结论是错误的.

2.6.2　模型的大小

在 2.5 节中证明紧致性定理的时候，我们是从和谐集 Γ 开始，构造一个可满足的结构 $\mathfrak{A}/E$. 这个结构有多大呢？其实，如果初始语言是可数的，那么 $|\mathfrak{A}/E|$ 就是可数的. 因此，和谐的句子集合在可数的语言中有可数的模型.

$\mathfrak{A}/E$ 是由初始结构 $\mathfrak{A}$ 开始构造的. $\mathfrak{A}$ 的论域是语言中的所有项，该语言则是添加了可数的新符号集的. 但是扩充后的语言仍是可数的，因此所有表达式的集合是可数的 (因此，所有项的集合也是可数的)，即 $|\mathfrak{A}|$ 是可数的.

$\mathfrak{A}/E$ 的论域包含了 $\mathfrak{A}$ 的元素的等价类，因此也是可数的集合. (可通过为每个等价类指派某个元素，建立从 $|\mathfrak{A}/E|$ 到 $|\mathfrak{A}|$ 中的一对一映射.）这样就有，$\mathfrak{A}/E$ 是可数的结构.

洛文海–斯科伦定理 (1915)　(a) 设 Γ 是可数语言中的可满足的公式集合，那么 Γ 在某个可数结构中是可满足的.

(b) 设 Σ 是可数语言中的句子集，如果 Σ 有任意模型，那么它有可数模型.

证明　首先，由可靠性定理，Γ 是和谐的. 然后由完备性定理 (及前面的说明)，Γ 可被可数结构满足. ■

(对该定理还有一个更直接的证明方法，该方法将会出现在 4.2 节及这一节的习题 1. 该证明不使用演绎计算，从一个任意的能够满足 Γ 的结构开始，通过各种操作从中提取一个能够满足 Γ 的可数 子结构.）

Γ 是一个句子情形下的洛文海 - 斯科伦 (Löwenheim-Skolem) 定理是洛文海 (Lepold Löwen heim) 于 1915 年发表的，斯科伦 (Thoralf Skolem) 在 1920 年将其扩充为可能无限的 Γ 的情形. 这个定理标志着数理逻辑进入了新的发展阶段. 早期的工作主要是利用形式语言及演绎计算进行数学形式化工作；这些工作主要是由弗雷格 (Gottlob Frege) 在 1879 年完成的. 例如，怀特黑德和罗素完成的《数学原理》(*Principia Mathematica*) (1910~1913) 对这个形式化工作进行了细化. 逻辑学的现代阶段始于逻辑学家返回去开始验证他们创建的形式系统. 该方向的其他早期工作是由希尔伯特、波斯特、哥德尔及塔斯基等人完成的.

151

我们来看一个洛文海 - 斯科伦定理应用的例子. 设 A_{ST} 是选定的集合论公理集合，我们当然希望这些公理是和谐的. 因此这些公理应该有一个模型. 根据洛文海 - 斯科伦定理，这些公理具有一个可数模型 $\mathfrak{C}$. 当然 $\mathfrak{C}$ 是所有由 A_{ST} 逻辑蕴涵的句子的模型. 其中一个句子断言，(根据指定的反映方式翻译回自然语言后) 存在不可数多个集合. 这里并没有矛盾，但是这种复杂的情况还是被含糊称为“斯科伦悖论”. 在结构$\mathfrak{C}$中，不存在从自然数到论域上的一对一映射的形式定义，这是毫无疑问的. 但是也无法排除 (在 $\mathfrak{C}$ 之外) 存在本征函数的可能性，这样的函数能够提供这样的一一对应.

结构 $\mathfrak{A}$ 的*理论*，记作 Th $\mathfrak{A}$，是 $\mathfrak{A}$ 中为真的所有句子的集合. 我们可以应用洛文海 - 斯科伦定理证明，对于可数语言 ($\Sigma = \mathrm{Th}\,\mathfrak{A}$) 的任意结构，存在可数的初等等价结构 $\mathfrak{B}$. 如果 $\mathfrak{B}$ 是 Th $\mathfrak{A}$ 的模型，那么 $\mathfrak{A} \equiv \mathfrak{B}$，因为

$$\models_{\mathfrak{A}} \sigma \Rightarrow \sigma \in \mathrm{Th}\,\mathfrak{A} \Rightarrow \models_{\mathfrak{B}} \sigma$$

及

$$\not\models_{\mathfrak{A}} \sigma \Rightarrow \models_{\mathfrak{A}} \neg\sigma \Rightarrow (\neg\sigma) \in \mathrm{Th}\,\mathfrak{A} \Rightarrow \models_{\mathfrak{B}} \neg\sigma \Rightarrow \not\models_{\mathfrak{B}} \sigma.$$

例如，实数域 $(\mathbb{R}; 0, 1+, \cdot)$ 是有限语言的不可数结构. 因此，肯定存在可数的结构 (也是域) 恰好满足同样的句子. (事实上，塔斯基证明我们可以用代数实数域取代它.)

例 考虑结构

$$\mathfrak{N} = (\mathbb{N}; 0, S, <, +, \cdot).$$

我们说存在可数结构 $\mathfrak{M}_0$，初等等价于 $\mathfrak{N}$(因此 $\mathfrak{M}_0$ 与 $\mathfrak{N}$ 恰好满足同样的句子)，但是不与 $\mathfrak{N}$ 同构.

证明 使用紧致性定理构造 $\mathfrak{M}_0$. 在语言中添加常数符号 c，设

$$\Sigma = \{\mathbf{0} < c, \mathbf{S0} < c, \mathbf{SS0} < c, \cdots\}.$$

152

我们说 $\Sigma \cup \mathrm{Th}\,\mathfrak{N}$ 有一个模型. 考虑它的一个有限子集，这个有限子集在如下结构中是真的：

$$\mathfrak{N}_k = (\mathbb{N}; 0, S, <, +, \cdot, k)$$

对某个很大的 k(其中 $k = c^{\mathfrak{N}_k}$). 根据紧致性定理，$\Sigma \cup \mathrm{Th}\,\mathfrak{N}$ 有一个模型.

由洛文海 - 斯科伦定理，$\Sigma \cup \mathrm{Th}\,\mathfrak{N}$ 有一个可数模型：

$$\mathfrak{M} = (|\mathfrak{M}|; \mathbf{0}^{\mathfrak{M}}, \mathbf{S}^{\mathfrak{M}}, <^{\mathfrak{M}}, +^{\mathfrak{M}}, \cdot^{\mathfrak{M}}, c^{\mathfrak{M}}).$$

设 $\mathfrak{M}_0$ 是在初始语言上对 $\mathfrak{M}$ 的限制：

$$\mathfrak{M}_0 = (|\mathfrak{M}|; \mathbf{0}^{\mathfrak{M}}, \mathbf{S}^{\mathfrak{M}}, <^{\mathfrak{M}}, +^{\mathfrak{M}}, \cdot^{\mathfrak{M}}).$$

由于 $\mathfrak{M}_0$ 是 Th $\mathfrak{N}$ 的模型，有 $\mathfrak{M}_0 \equiv \mathfrak{N}$.

$$\models_{\mathfrak{N}} \sigma \Rightarrow \sigma \in \text{Th } \mathfrak{N} \Rightarrow \models_{\mathfrak{M}_0} \sigma$$
$$\not\models_{\mathfrak{N}} \sigma \Rightarrow \neg\, \sigma \in \text{Th } \mathfrak{N} \Rightarrow \models_{\mathfrak{M}_0} \neg\, \sigma \Rightarrow \not\models_{\mathfrak{M}_0} \sigma.$$

至于可数结构 $\mathfrak{M}_0$ 不与 $\mathfrak{N}$ 同构的证明，留给读者作练习．($|\mathfrak{M}_0|$ 包含无限大的数 $c^{\mathfrak{M}}$)　■

不可数[1] 语言的情况如何呢？假如在完备性定理的证明中，我们从基数为 λ 的语言中的集合 Γ 开始．我们说在这种情况下，结构 $\mathfrak{A}/E$ 的基数 $\leqslant \lambda$.

$\mathfrak{A}/E$ 是由初等结构 $\mathfrak{A}$ 构造来的．其论域是在语言中加入 λ 个常数符号后得到的所有项的集合．因此，扩充语言仍具有基数 λ. 因此 (根据定理 0D) 所有表达式的集合的基数 $\leqslant \lambda$(因而所有项的集合也是这样)．(事实上，至少有 λ 个新常数符号，项集合的基数恰好是 λ.)

$\mathfrak{A}/E$ 的论域包含 $\mathfrak{A}$ 中元素的等价类，因此 card $|\mathfrak{A}/E| \leqslant$ card $|\mathfrak{A}|$. (可以通过为每个等价类指定某个选定的元素来建立由 $|\mathfrak{A}/E|$ 到 card $|\mathfrak{A}|$ 中的一对一映射，然而这里我们使用选择公理．) 这样，Γ 在基数 $\leqslant \lambda$ 的结构 $\mathfrak{A}/E$ 中是能够满足的．

洛文海–斯科伦定理　(a) 设 Γ 是基数为 λ 的语言中可满足的公式集合，那么 Γ 在某个基数小于 λ 的结构中是可满足的．

(b) 设 Σ 是基数为 λ 的语言中的句子集，如果 Σ 有任意模型，那么它有基数小于等于 λ 的模型．

153

先前的洛文海 – 斯科伦定理是这个定理的特例，那里的 $\lambda = \aleph_0$.

设有一个可数语言的不可数结构 $\mathfrak{A}$，根据洛文海 – 斯科伦定理 (应用到 Th $\mathfrak{A}$ 上)，存在可数的 $\mathfrak{B}$，它是 Th $\mathfrak{A}$ 的模型，因此，如前所述，$\mathfrak{A} \equiv \mathfrak{B}$.

反之，假如从可数结构 $\mathfrak{B}$ 开始，是否存在不可数的 $\mathfrak{B}$ 使得 $\mathfrak{A} \equiv \mathfrak{B}$? 如果 $\mathfrak{B}$ 是有限的 (且语言包含等号)，那么这是不可能的．但是如果 $\mathfrak{B}$ 是无限的，那么根据 “升降洛文海 – 斯科伦定理”，存在这样的 $\mathfrak{A}$. 升洛文海 – 斯科伦定理归功于塔斯基 (Tarski)，即 “LST” 中的 “T” .

LST 定理　设 Γ 是基数为 λ 的语言中可满足的公式集合，且假定其在某个无限结构中是可满足的，那么对每个基数 $\kappa \geqslant \lambda$，存在基数为 κ 的结构，在其中 Γ 是可满足的．

证明　设 $\mathfrak{A}$ 是满足 Γ 的无限结构．通过添加基数为 κ 的新常数符号集合 C 对语言进行扩充．设

$$\Sigma = \{c_1 \neq c_2 | c_1, c_2 \text{ 是}C\text{的不同的元素}\}.$$

在结构 $\mathfrak{A}$ 中，$\Sigma \cup \Gamma$ 的任意有限子集都是可满足的，为所取有限子集中的有限多个常数符号指定互不相同的对象即可．(由于 $\mathfrak{A}$ 是无限的，所以足以容纳任意有限多个对象．) 因此由紧致性定理，$\Sigma \cup \Gamma$ 是可满足的，再由洛文海 – 斯科伦定理，其在基数 $\leqslant \kappa$ 的结构 $\mathfrak{B}$ 中是可满足的．(扩充语言的基数为 $\kappa + \lambda = \kappa$.) 但 Σ 的任何一个模型的基数显然都 $\geqslant \kappa$. 因此，$\mathfrak{B}$ 的基数必定是 κ；再将 $\mathfrak{B}$ 限制到初始语言上．　■

---

1. 想避开不可数基数的读者可以跳到 2.6.3 节．

推论 26F (a) 设 Σ 是可数语言中的句子集合，如果 Σ 有某个无限模型，那么 Σ 有任意大无限基数的模型.

(b) 设 $\mathfrak{A}$ 是可数语言中的无限结构，对任意无穷基数 λ，存在基数为 λ 的结构 $\mathfrak{B}$ 使得 $\mathfrak{A} \equiv \mathfrak{B}$.

证明 (a) 在定理中取 $\Gamma = \Sigma, \lambda = \aleph_0$ 即得.

(b) 在 (a) 中取 $\Sigma = \text{Th}\ \mathfrak{A}$. ■

考虑非逻辑公理的句子集合. (如 Σ 为集合论的公理集或者数论的公理集.) 称 Σ 是 *范畴的* 当且仅当 Σ 的任意两个模型都是等价的. 上述推论意味着如果 Σ 有任意大无限模型，那么 Σ 不是范畴的. 例如不存在这样的句子集合，其模型恰好是同构于 $(\mathbb{N}; 0, S, +, \cdot)$ 的结构. 这是对一阶语言表达能力的局限性的陈述. (在 4.1 节，我们会看到存在范畴的二阶句子. 但是二阶句子是特殊的，可以对 *子集* 加以描述，避免由结构给出不同的解释.)

154

2.6.3 理论

理论 定义为逻辑蕴涵意义下封闭的句子集合，也就是说，T 是一个理论当且仅当 T 是一些句子的集合，该集合使得语言中的任意句子 σ，

$$T \vDash \sigma \Rightarrow \sigma \in T.$$

(注意：我们仅指句子，而不是带有自由变量的公式.)

例如，总存在仅包含语言的恒真句子的最小的理论. 另外，存在包含语言的所有句子的理论；这是唯一不能够满足的理论.

对 (语言的) 结构类 $\mathcal{K}$，定义其 *理论* 为 (记作 Th $\mathcal{K}$)

$$\text{Th}\ \mathcal{K} = \{\sigma | \sigma \text{ 在}\mathcal{K}\text{的每个元素中都是真的}\}$$

(这一概念由先前的特殊情况 $\mathcal{K} = \{\mathfrak{A}\}$ 而来.)

定理 26G Th $\mathcal{K}$ 确实是一个理论.

证明 $\mathcal{K}$ 的任何元素都是 Th $\mathcal{K}$ 的模型. 这样，如果 σ 在 Th $\mathcal{K}$ 的每个模型中都是真的，那么它在 $\mathcal{K}$ 的每个结构中都是真的. 因而，它属于 Th $\mathcal{K}$. ■

例如，如果语言的参数包括 $\forall$, $\mathbf{0}$, $\mathbf{1}$, $+$ 和 $\cdot$，且 $\mathcal{F}$ 是所有域的类，那么域的理论 Th $\mathcal{F}$ 是语言在任意域中都真的的所有句子的集合. 如果 $\mathcal{F}_0$ 是特征为 0 的域的类，那么 Th $\mathcal{F}_0$ 是特征为 0 的域的理论.

回忆一下，对于句子集合 Σ，定义 Mod Σ 为 Σ 的所有模型的类. Th Mod Σ 则是在 Σ 的所有模型中都是真的的所有句子的集合. 但是这不过是由 Σ 逻辑蕴涵的所有句子的集合. 称此集合为 Σ 的 *推论集*，记作 CnΣ. 这样，

$$\begin{aligned}\text{Cn}\Sigma &= \{\sigma | \Sigma \vDash \sigma\} \\ &= \text{Th Mod } \Sigma.\end{aligned}$$

例如，集合论是特定的句子集合的推论集，这些句子无疑就是集合论的公理. 一个句子集合 T 是一个理论当且仅当 $T = \text{Cn}T$.

155

一个理论 T 称为是 *完备的* 当且仅当对每个句子 σ，或者 $\sigma \in T$ 或者 $(\neg\, \sigma) \in T$. 例如，对于任意结构 $\mathfrak{A}$，$\text{Th}\,\{\mathfrak{A}\}$(如前可记作 $\text{Th}\,\mathfrak{A}$) 总是一个完备的理论. 事实上，很明显，$\text{Th}\mathcal{K}$ 是完备的理论当且仅当 $\mathcal{K}$ 的任意两个结构都是初等等价的. 一个理论 T 是完备的当且仅当 T 的任意两个模型是初等等价的.

例如，域的理论是不完备的，因为句子

$$\mathbf{1} + \mathbf{1} = \mathbf{0},$$

$$\exists x\, x \cdot x = \mathbf{1} + \mathbf{1}$$

在某些域中是真的而在另外一些域中是假的. 特征为 0 的代数闭域的理论是完备的，但这并不明显 (见定理 26J).

***定义**　理论 T 是 *可公理化的* 当且仅当存在可判定的句子集合 Σ 使得 $T = \text{Cn}\Sigma$.

定义　理论 T 是 *有限可公理化的* 当且仅当对某个有限句子集合 Σ, $T = \text{Cn}\Sigma$.

在后一种情况中，我们有 $T = \text{Cn}\{\sigma\}$(记作：$T = \text{Cn}\sigma$)，其中 σ 是 Σ 的有限多个元素的合取. 比如，域的理论是有限可公理化的. 对于域的类 $\mathcal{F}$ 是 $\text{Mod}\ \Phi$，其中 Φ 是域公理的有限集合. 域的理论是 $\text{Th Mod}\Phi = \text{Cn}\Phi$.

特征为 0 的域的理论是可公理化的，记作 $\text{Cn}\Phi_0$，其中 Φ_0 包含 (有限多个) 域公理以及无限多个句子：

$$\mathbf{1} + \mathbf{1} \neq \mathbf{0},$$

$$\mathbf{1} + \mathbf{1} + \mathbf{1} \neq \mathbf{0},$$

$$\cdots$$

这个理论不是有限可公理化的. 为了证明这一点，首先要注意到 Φ_0 没有一个有限子集具有完整的理论作为其推论集. (由于有限子集可能在特征非常大的某个域中是真的.) 应用如下定理：

定理 26H　如果 $\text{Cn}\Sigma$ 是有限可公理化的，那么存在有限的 $\Sigma_0 \subseteq \Sigma$ 使得 $\text{Cn}\Sigma_0 = \text{Cn}\Sigma$.

证明　假定 $\text{Cn}\Sigma$ 是有限可公理化的，那么对某个句子 τ，$\text{Cn}\Sigma = \text{Cn}\tau$. 通常，$\tau \notin \Sigma$，但是至少会有 $\Sigma \vDash \tau$. ($\tau \in \text{Cn}\tau = \text{Cn}\Sigma$.) 根据紧致性定理，存在一个有限的 $\Sigma_0 \subseteq \Sigma$ 使得 $\Sigma_0 \vDash \tau$，那么

$$\text{Cn}\tau \subseteq \text{Cn}\Sigma_0 \subseteq \text{Cn}\Sigma,$$

156

因此等号就成立了. ■

下面使用目前的术语重新叙述推论 25F 和 25G.

推论 26I　(a) (在合理的语言中) 一个公理化的理论是能行可枚举的.
(b) (在合理的语言中) 完备的可公理化的理论是可判定的.

我们可以通过如下的交换图来表达这些概念之间的关系 (其中已经包含了习题 6 的结果).



例如, 以公理形式给出的理论 (比如策梅洛 – 弗兰克尔集合论, 即对某个特定集合 A_{ZF} 而言的 $\mathrm{Cn}A_{\mathrm{ZF}}$) 是有限可枚举的. 3.7 节会证明集合论是不可判定的也是不完备的. 数论, 即结构 $(\mathbb{N}; 0, S, <, +, \cdot, E)$ 的理论是完备的但不是能行可枚举的, 因此也不是可公理化的 (3.5 节).

只要我们能够证明所涉及的理论是完备的, 就可使用上述推论的 (b) 来建立公理化理论的可判定性. 有时, 可以使用洛斯 – 瓦特测试来检查完备性.

称理论 T 是 $\aleph_0$ 范畴的当且仅当 T 的可数无限模型都是同构的. 更一般地说, 对于基数 κ, 称 T 是 κ 范畴的当且仅当 T 的所有基数为 κ 的模型都是同构的.

洛斯–瓦特测试 (1954) 设 T 是可数语言的理论, 假设 T 没有有限模型,

(a) 如果 T 是 $\aleph_0$ 范畴的, 那么 T 是完备的.

(b) 对某个无限基数 κ, 如果 T 是 κ 范畴的, 那么 T 是完备的.

证明 可以证明对于 T 的任意两个模型 $\mathfrak{A}$ 和 $\mathfrak{B}$, 有 $\mathfrak{A} \equiv \mathfrak{B}$. 因为 $\mathfrak{A}$ 和 $\mathfrak{B}$, 是无限的, 那么存在基数为 κ 的结构 $\mathfrak{A}' \equiv \mathfrak{A}$, $\mathfrak{B}' \equiv \mathfrak{B}$. $\mathfrak{A}'$ 同构于 $\mathfrak{B}'$, 因此我们有

$$\mathfrak{A} \equiv \mathfrak{A}' \cong \mathfrak{B}' \equiv \mathfrak{B}.$$

即 $\mathfrak{A} \equiv \mathfrak{B}$. ■

(如果 T 是基数为 λ 的语言中的理论, 那么我们必须要求 $\lambda \leqslant \kappa$.) 157

洛斯 – 瓦特测试反过来是不成立的. 即对于任意的 κ, 存在非 κ 范畴的完备理论.

在 3.1 节, 我们应用洛斯 – 瓦特测试来证明带有 0 及后继数的自然数理论的可判定性. 它也可用于证明复数域的可判定性. 但是这个证明完全是代数的.

定理 26J (a) 特征为 0 的代数闭域的理论是完备的.

*(b) 复数域

$$\mathfrak{C} = (\mathbb{C}; 0, 1, +, \cdot)$$

的理论是可判定的.

证明 设 $\mathcal{A}$ 是特征为 0 的代数闭域的类, 那么 $\mathcal{A} = \mathrm{Mod}(\Phi_0 \cup \Gamma)$, 其中 Φ_0 包含特征为 0 的域的公理, 且 Γ 包含句子:

$$\forall a \forall b \forall c (a \neq \mathbf{0} \to \exists x a \cdot x \cdot x + b \cdot x + c = \mathbf{0}),$$

$$\forall a \forall b \forall c \forall d (a \neq \mathbf{0} \to \exists x a \cdot x \cdot x \cdot x + b \cdot x \cdot x + c \cdot x + d = \mathbf{0}),$$

$$\cdots$$

集合 $(\phi_0 \cup \Gamma)$ 是可判定的且 $\mathrm{Th}\mathcal{A} = \mathrm{Cn}(\Phi_0 \cup \Gamma)$，因此这个理论是可公理化的. (a) 说明理论是完备的，因此也是可判定的.

(b) 可从 (a) 得到. 我们有 $\mathfrak{C} \in \mathcal{A}$，因此 $\mathrm{Th}\mathcal{A} \subseteq \mathrm{Th}\mathfrak{C}$，$\mathrm{Th}\mathcal{A}$ 的完备性意味着等号成立，见习题 2.

为证明 (a)，应用洛斯 – 瓦特测试. $\mathrm{Th}\mathcal{A}$ 的模型恰好是 $\mathcal{A}$ 的元素，这些都是无限的. 进一步断定 $\mathrm{Th}\mathcal{A}$ 在不可数基数下是范畴的. 这等价于说任何两个具有相同的不可数基数的特征为 0 的代数闭域是同构的.

最后的断定是代数上著名的结果，这里略去其证明. 任何一个域 $\mathfrak{F}$ 可通过如下方式获得：(1) 由素的子域开始，其能够由特征 $\mathfrak{F}$ 在同构范围内确定；(2) 可以做一个超越扩充，这可以通过超越基的基数在同构范围内确定，即由 $\mathfrak{F}$ 的超越度确定 (在其素子域之上)；(3) 最后，做代数扩充. 这样就有一个 Steinitz 定理：两个代数闭域是同构的当且仅当它们有相同的特征和相同的超越度.

158

如果一个无限域 $\mathfrak{F}$ 的超越度为 κ，那么 $\mathfrak{F}$ 的基数是 κ 与 $\aleph_0$ 中较大的一个. 因此，对不可数域，基数等于超越度. 因此，从 Steintz 定理我们得到，两个具有相同特征和相同不可数基数的代数闭域是同构的. ■

实数域 $(\mathbb{R}; 0, 1, +, \cdot)$ 的理论也是可判定的. 但是这一结果 (归功于塔斯基) 比上述定理还要深一些. 在任何无限基数下，实数域的理论都不是范畴的，因此不能应用洛斯 – 瓦特测试.

作为最后的一个应用，可以证明有理数序初等等价于实数序，即

$$(\mathbb{Q}; <_Q) \equiv (\mathbb{R}; <_R),$$

其中 $\mathbb{Q}$ 与 $\mathbb{R}$ 分别是有理数和实数，而 $<_Q$ 和 $<_R$ 则是相应的序. 为证明它们的初等等价性，需要验证它们都是完备理论的模型 (其必定与每个结构的理论相一致). 康托尔定理提供了其中的关键要素：任何两个可数的无端点的稠密线性序都是等价的.

为给出证明的详细过程，这里我们从语言的讨论开始. 语言要包括等号及参数 $\forall$ 和 $<$，设 δ 是下述句子的合取：

(1) 序公理 (三分律及传递性)：

$$\forall x \forall y (x < y \vee x = y \vee y < x),$$
$$\forall x \forall y (x < y \to y \not< x),$$
$$\forall x \forall y \forall z (x < y \to y < z \to x < z),$$

(2) 稠密性：

$$\forall x \forall y (x < y \to \exists z (x < z < y)).$$

(3) 无端点：

$$\forall x \exists y \exists z (y < x < z).$$

根据定义稠密的无端点的线性序是该语言的结构，即 δ 的模型，很明显这都是无限的. 更进一步，我们说这些序的理论，$\mathrm{Cn}\delta$，是 $\aleph_0$ 范畴的. 这由如下定理而来.

定理 26K (康托尔定理) 任何 δ 的可数模型都同构于 $(\mathbb{Q}; <_Q)$.

其证明留作习题 (习题 4).

159

可以应用洛斯 - 瓦特测试验证 $\mathrm{Cn}\delta$ 是完备的，因此 δ 的任意两个模型都是初等等价的，特别地，

$$(\mathbb{Q}; <_Q) \equiv (\mathbb{R}; <_R).$$

我们也可以证明这些结构都有可判定的理论.

2.6.4 前束范式

有时将公式中的量词符号全部移到其他符号的左边更方便些. 例如,

$$\forall x(Ax \to \forall y Bxy)$$

等价于

$$\forall x \forall y(Ax \to Bxy).$$

且

$$\forall x(Ax \to \exists y Bxy)$$

等价于

$$\forall x \exists y(Ax \to Bxy).$$

前束 定义为下述形式，对某个 $n \geqslant 0$,

$$Q_1 x_1 \cdots Q_n x_n \alpha,$$

其中 Q_i 是 $\forall$ 或者 $\exists$, α 是无量词的表达式.

前束范式定理 对任意公式，都可以找到一个逻辑等价的前束范式.

证明 可以使用下述的量词操作规则:

Q1a.	$\neg \forall x \alpha \models\!\!\!\dashv \exists x \neg \alpha$.	
Q1b.	$\neg \exists x \alpha \models\!\!\!\dashv \forall x \neg \alpha$.	
Q2a.	$(\alpha \to \forall x \beta) \models\!\!\!\dashv \forall x(\alpha \to \beta)$	对 α 中的非自由变量 x.
Q2b.	$(\alpha \to \exists x \beta) \models\!\!\!\dashv \exists x(\alpha \to \beta)$	对 α 中的非自由变量 x.
Q3a.	$(\forall x \alpha \to \beta) \models\!\!\!\dashv \exists x(\alpha \to \beta)$	对 β 中的非自由变量 x.
Q3b.	$(\exists x \alpha \to \beta) \models\!\!\!\dashv \forall x(\alpha \to \beta)$	对 β 中的非自由变量 x.

Q1 是显然的，其他的参见 2.4 节及其习题 8.

可以使用归纳法证明每个公式都有等价的前束公式.

(1) 对原子公式显然没问题，无量词的公式显然也是前束公式.

(2) 如果 α 等价于前束公式 α', 那么 $\forall x \alpha$ 等价于 $\forall x \alpha'$.

160

(3) 如果 α 等价于前束公式 α', 那么 $\neg \alpha$ 等价于 $\neg \alpha'$. 对 $\neg \alpha'$ 应用 Q1 可得前束公式；例如,

$$\neg \forall x \exists y \exists z \beta \models\!\!\!\dashv \exists x \forall y \forall z \neg \beta.$$

(4) 最后考虑 $\alpha \to \beta$ 的情形. 由归纳假设, 对 α 与 β 有等价的公式 α' 与 β'. 由字母变换式的定理, 可进一步假设在 α' 与 β' 中约束出现的任何变量都不在另外一个中出现. 使用 Q2 及 Q3 可以得到等价于 $\alpha' \to \beta'$ 的前束公式 (自然也等价于 $\alpha \to \beta$). 注意到存在某个范围可使用规则 Q2 及 Q3. 例如,

$$\forall x \exists y \varphi \to \exists u \psi$$

(其中 x 和 y 都不在 ψ 中自由出现, u 不在 φ 中自由出现) 等价于下面的任意一个:

$$\exists x \forall y \exists u(\varphi \to \psi),$$
$$\exists x \exists u \forall y(\varphi \to \psi),$$
$$\exists u \exists x \forall y(\varphi \to \psi).$$
■

2.6.5 注记

本书一开始就声明符号逻辑是演绎推理的数学模型. 由于我们学习的知识已经足够多, 该是考虑一下这个声明的时候了.

作为第一个例子, 考虑研究集合论的数学家, 使用带有等号、属于符号 $\in$ 及其他符号 ($\varnothing$、$\cup$ 等) 的语言. 原则上, 这些后定义的符号可以消去, 任意句子可以由等价的不出现这些后定义符号的公式代替 (见 2.7 节). 他将集合及属于作为基本 (或者未定义的) 概念, 采用某个包含这些概念的公理集合 A_{ST}, 断定对于某些特定句子 (定理), 无论这些未定义的概念实际上的含义是什么, 只要公理是真的, 这些句子就是真的. 为了支持这些断定, 他给出证明, 这些证明都是有限长度的论证, 以向其同事证实这些断定的正确性.

用一阶逻辑的术语, 我们可以将上述过程描述如下: 语言即带有等号和二元谓词符号 $\in$, 这样 $\forall$ 与 $\in$ 仅是参数而已. 这个语言中存在某个句子集合 A_{ST} 列出(非逻辑) 的公理集. 那么, 其他句子是 A_{ST} 的逻辑推论, 即在 A_{ST} 的任意模型中都是真的. 如果 τ 是 A_{ST} 的推论 (仅当这种情况), 存在从 A_{ST} 到 τ 的演绎.

161

接下来, 考虑更典型的数学家, 即代数学家或者分析家的情况. 代数学家使用群论的公理, 但是他也研究某些集合论的内容. 类似地, 分析家处理数论与集合论的相关的句子. 在这两种情况下, 原则上, 都可以将代数和分析中的断言转换为集合论中的断言. 那么上一段的评论在这里也可使用.

对数学家而言, 符号逻辑的意义主要在于其能够精确反映数学推理. 符号逻辑与非符号逻辑构成了哲学研究的一个传统领域, 即研究人们获得特定观点的过程. 一阶逻辑应用的非数学的例子存在于很多无关紧要的情形中, Lewis Carrol 给出了一些例子, 其中一个是这样的: 可以从如下 3 个条件下得到婴儿不能够对付鳄鱼. 这 3 个假设条件是: (1) 婴儿是不合逻辑的; (2) 能够对付鳄鱼的人是不可轻视的; (3) 不合逻辑的人是可以轻视的.

可是, 并非无关紧要的情况如何呢? 由于我们通常无法显式地描述推理的假设条件, 这种情况下的应用是难以说清的. 当然, 在某些特定的领域 (诸如物理学、医学及法学等各种领域) 中, 假设条件不仅可以显式地表达, 而且正在显式地表达. 在某些情况下, 现实生活的形式化推理并不需要一阶逻辑的全部功能, 而在另外一些情况下——从日常生活到量子力学——也许会需要的更多.

习题

1. 证明下述句子是有限恒真的 (即在每个有限结构中是真的):
   (a) $\exists x\exists y\exists z[(Pxfx \to Pxx) \vee (Pxy \wedge Pyz \wedge \neg\, Pxz)]$
   (a) $\exists x\forall y\exists z[(Qzx \to Qzy) \to (Qxy \to Qxx)]$
   提示: 这些句子的逆的任意模型都是无限的.
2. 设 T_1 和 T_2 是 (同一语言的) 理论, 使得 (i) $T_1 \subseteq T_2$, (ii) T_1 是完备的; (iii) T_2 是可满足的. 证明: $T_1 = T_2$. 162
3. 判定以下事实:
   (a) $\Sigma_1 \subseteq \Sigma_2 \Rightarrow \text{Mod } \Sigma_2 \subseteq \text{Mod } \Sigma_1$.
   $\mathcal{K}_1 \subseteq \mathcal{K}_2 \Rightarrow \text{Th } \mathcal{K}_2 \subseteq \text{Th } \mathcal{K}_1$.
   (b) $\Sigma \subseteq \text{Th Mod } \Sigma$ 且 $\mathcal{K} \subseteq \text{Mod Th } \mathcal{K}$
   (c) $\text{Mod } \Sigma = \text{Mod Th Mod } \Sigma$ 且 $\text{Th } \mathcal{K} = \text{Th Mod Th } \mathcal{K}$. (c 可以从 (a) 和 (b) 得到.)
4. 证明: 任意两个无端点的可数稠密线性序是同构的 (定理 26K). **提示**: 设 $\mathfrak{A}$ 与 $\mathfrak{B}$ 是两个结构, $|\mathfrak{A}| = \{a_0, a_1, \cdots\}$, $|\mathfrak{B}| = \{b_0, b_1, \cdots\}$, 逐步构造一个同构; 在第 $2n$ 步保证 a_n 与某个合适的 b_j 匹配, 在第 $2n+1$ 步, 保证 b_n 与某个合适的 a_i 匹配.
5. 给出与如下公式等价的前束公式:
   (a) $(\exists x Ax \wedge \exists x Bx) \to Cx$.
   (b) $\forall x Ax \leftrightarrow \exists x Bx$.

*6. 证明推论 26I 的 (a) 的逆: (在合适的语言中) 能行可枚举的理论是可公理化的. **提示**: 集合 $\{\sigma_0, \sigma_1, \sigma_2, \cdots\}$(在具有相同模型的意义下) 等价于集合 $\{\sigma_0, \sigma_0 \wedge \sigma_1, \sigma_0 \wedge \sigma_1 \wedge \sigma_2, \cdots\}$.

7. 考虑带有二元谓词符号 $<$ 的语言, 设 $\mathfrak{N} = (\mathbb{N}; <)$ 是包含 (通常序的) 自然数的结构. 证明: 存在某个初等等价于 $\mathfrak{N}$ 的 $\mathfrak{A}$, 使得 $<^{\mathfrak{A}}$ 具有降序链. (即在 $|\mathfrak{A}|$ 中存在 $a_0, a_1, \cdots$, 使得对所有 i 有 $\langle a_{i+1}, a_i\rangle \in <^{\mathfrak{A}}$.) **提示**: 应用紧致性定理.
   说明: 该习题的目的在于说明在这个语言中无法表达 "不存在降序链. "
8. 设 σ 在理论 T 的所有无限模型中都是真的, 证明存在有限数字 k 使得 σ 在 T 的所有模型 $\mathfrak{A}$ 中都是真的, 这里 $|\mathfrak{A}|$ 具有 k 个或者更多个元素.
9. 称句子集 Σ 具有 *有限模型性质* 当且仅当对于 Σ 的每个元素 σ, 如果有任意模型就有有限模型. 设 Σ 是有限语言 (即具有有限多个参数的语言) 中的句子集, 且 Σ 具有有限模型性质. 对给定的判定过程及 Σ 的任意元素 σ, 能否判定 σ 是否具有任意模型. **提示**: 这样的句子集是否是有限可枚举? 其补是否是有限可枚举?
10. 设有一个不带函数符号的有限语言,
    (a) 证明: 可满足的 $\exists_2$ 句子集是可判定的 (术语及相关背景见 2.2 节的习题 19). **提示**: 应用上一习题. 163
    (b) 证明: 恒真的 $\forall_2$ 句子集是可判定的. ($\forall_2$ 公式是指具有形式 $\forall x_1 \cdots \forall x_m \exists y_1 \cdots \exists y_n \theta$ 的公式, 其中 θ 是无量词的公式.)

    说明: 在一阶逻辑中, "判定问题" (Entscheidungs 问题) 就是在给定一个公式后判定其是否恒真. 由丘奇定理 (3.5 节), 这一问题通常是无解的. 这个习题给出的是判定问题中的一个可解的情形.

2.7 理论之间的解释[1]

在某些情况下, 理论 T_1 完全可以表达和另一个理论 T_0 相同的内容. 一个典型的例子就是这两个理论都在相同的语言中, 并且 $T_0 \subseteq T_1$. 但如果两个理论分别在不同的语言中, 应

1. 本节内容仅在 3.7 节中使用.

该存在一种方法，将一种语言中 T_0 的元素翻译成另一种语言中 T_1 的元素. 我们将在本节中讨论这个问题.

首先，我们讨论符号的问题，这个问题可以作为上一段讨论的问题的例子，其中，T_0 是通过对 T_1 添加新符号而得到的. 如果我们已经适当地给出定义，那么最初的理论原则上应该和新的理论一样“强”. 我们只考虑函数符号的情况，因为谓词符号的情况和函数符号没有本质的区别.

2.7.1 定义函数

在数学中，一个新函数的定义通常是非常有用的. 例如，在集合论中幂集运算符 $\mathcal{P}$ 用下面的句子来定义，“设 $\mathcal{P}x$ 表示一个集合，它的元素都是 x 的子集. ” 或者用形式化的语言表示为 (语言中包含 $\in$, $\subseteq$ 和 $\mathcal{P}$),

$$\forall v_1 \forall v_2 [\mathcal{P} v_1 = v_2 \leftrightarrow \forall u (u \in v_2 \leftrightarrow u \subseteq v_1)].$$

我们这里所谈论的定义和定理、公理不同. 和定理不同的是，定义不需要证明，只要陈述出来就行了. 和公理不同的是，我们不能希望定义能够给出一些实质的信息，它只能为我们提供一些使用上的方便，而不是新的知识.

164

如果这个希望能够实现，那么定义必须通过一个合理的方式给出. 数论中有一个最不合理的例子，假设我们通过“定义”引进一个新的函数

$$f(x) = y \text{ iff } x < y.$$

（或者用形式化语言中的句子表示为 $\forall v_1 \forall v_2 (f v_1 = v_2 \leftrightarrow v_1 < v_2)$. ）由于 $1 < 2$，我们有 $f(1) = 2$. 但又由于 $1 < 3$，所以有 $f(1) = 3$. 因此我们就能得出结论 (结论本身不包含 f)：$2 = 3$.

显然，f 的这个定义从某种意义上说是很差的，它不仅没有给我们提供方便，还使我们得出 $2 = 3$ 的结果，而这个结果在没有这个定义的情况下是无法得出的. 这个定义还把“$f(1)$”和其他的数 (2, 3 等) 混淆起来了，这样，$f(1)$ 就没有“明确定义”. 一个名称必须指派给唯一一个对象.

在这一小节中，我们要讨论在什么样的情况下定义是令人“满意”的. 为了简单起见，我们考虑只定义一元函数符号 f. 当然对于多元函数符号来说也是对的.

现在我们考虑不包含一元函数符号的 f 一个语言上的理论 T. (例如，T 可以是集合论中公理的推论集合.) 我们通过下面的 f 定义，把它加到语言中：

$$\forall v_1 \forall v_2 [f v_1 = v_2 \leftrightarrow \varphi], \tag{δ}$$

其中 φ 是原始语言中的公式 (即不包含 f 的公式)，φ 中只有 v_1 和 v_2 是自由出现的.

定理 27A 在上述情况下，下面的结果是等价的：

(a) (f 的定义没有创新.) 对于较小语言中的任意一个句子 σ，如果 (在讨论的语言中)，$T;\delta \vDash \sigma$，那么就有 $T \vDash \sigma$.

(b) (f 是明确定义的.) 句子

$$\forall v_1 \exists! v_2 \varphi \tag{ε}$$

在理论 T 中. (这里 "$\exists! v_2\varphi$" 是一个长公式的缩写, 具体见 2.2 节的习题 21.)

证明 要证明 (a)⇒(b), 只要注意到 $\delta \models \varepsilon$ 就可以了. 因此在 (a) 中取 $\sigma = \varepsilon$, 我们就能得到 $T \models \varepsilon$.

反过来, 假设 $T \models \varepsilon$. 令 $\mathfrak{A}$ 是 T 的模型($\mathfrak{A}$ 是原始语言中的结构). 对于 $d \in |\mathfrak{A}|$, 令 $F(d)$ 是唯一的 $e \in |\mathfrak{A}|$ 满足 $\models_{\mathfrak{A}} \varphi[\![d, e]\!]$. (这样唯一的 e 是存在的, 因为 $\models_{\mathfrak{A}} \varepsilon$.) 令 $(\mathfrak{A}, F)$ 是我们要讨论的语言中的结构, 并且 $\mathfrak{A}$ 在原始参数上, F 被指派给符号 f. 易见, $(\mathfrak{A}, F)$ 是 δ 的模型, 而且 $\mathfrak{A}$ 和 $(\mathfrak{A}, F)$ 满足原始语言中相同的句子. 特别地, $(\mathfrak{A}, F)$ 是 T 的模型. 因此,

165

$$\begin{aligned} T; \delta \models \sigma &\Rightarrow \models_{(\mathfrak{A}, F)} \sigma \\ &\Rightarrow \models_{\mathfrak{A}} \sigma. \end{aligned}$$

■

(我们可以用二阶逻辑使上面的讨论过程更简洁些, ε 逻辑等价于句子 $\exists f \delta$.)

2.7.2 解释

本小节的基本思想是, 一个理论有可能和另一个语言中的理论一样 "强" (严格意义上说). 当我们同时考虑两个语言时, 它们两者应该不会冲突, 比如一个语言中的否定符号不能是另一个语言中的谓词符号. 我们可以假设每个语言都是从另一个父辈语言中去掉一些参数后得到的 (有可能去掉等号), 那么, 这种冲突就是可以避免的.

例如, 公理集合论至少与含有零和后继数的自然数理论 (即 $(\mathbb{N}; 0, S)$ 的理论) 一样强. $(\mathbb{N}; 0, S)$ 语言中的任意一个句子都可以用自然的方式翻译成集合论中的句子. (这种翻译方法在 3.7 节中有简略的介绍.) 如果原始句子在 $(\mathbb{N}; 0, S)$ 中取值为真, 那么翻译后的句子将是集合论公理的推论. (这并不显然, 证明的过程中要使用 3.1 节中的结论.)

下面我们仔细地研究一下第 2 个例子. 一方面, 考虑理论

$$(\mathbb{N}; 0, S)$$

另一方面考虑理论

$$(\mathbb{Z}; +, \cdot)$$

(此处 $\mathbb{Z}$ 表示所有整数, 包括正数、负数和零.) 我们断言第二个理论和第一个理论一样强. 那么, 含有 0 和 S 的自然数语言 $\mathbb{N}$ 中的一个句子如何翻译成含有加法和乘法的整数 $\mathbb{Z}$ 中的句子呢?

数论中的拉格朗日定理为我们提供了第一个线索: 一个整数是非负的当且仅当它是 4 个平方数的和. 那么第一个语言中的量词 $\forall x$(其中 x 在 $\mathbb{N}$ 中) 可以用第二个语言中的下列公式来代替:

166

$$\forall x(\exists y_1 \exists y_2 \exists y_3 \exists y_4 x = y_1 \cdot y_1 + y_2 \cdot y_2 + y_3 \cdot y_3 + y_4 \cdot y_4 \rightarrow)$$

第二个线索是 $\{0\}$ 和后继函数 (看作关系) 在 $(\mathbb{Z}; +, \cdot)$ 中是可定义的. 集合 $\{0\}$ 可定义为

$$v_1 + v_1 = v_1.$$

($\mathbb{Z}$ 中的) 后继关系可以定义为

$$\forall z(z \cdot z = z \wedge z + z \neq z \rightarrow v_1 + z = v_2).$$

那么 $(\mathbb{N};0,S)$ 中的句子

$$\forall x \mathbf{S} x \neq \mathbf{0}$$

可以译作

$$\begin{aligned}&\forall x[\exists y_1\exists y_2\exists y_3\exists y_4 x = y_1\cdot y_1 + y_2\cdot y_2 + y_3\cdot y_3 + y_4\cdot y_4 \to \\ &\neg\,\forall u(u+u=u\to\forall v(\forall z(z\cdot z = z\wedge z+z\neq z\to x+z=v)\to v=u))].\end{aligned}$$

例子就到此为止．为了一般性的讨论，我们引入下面的记号

$$\varphi(t)=\varphi_t^{v_1},$$
$$\varphi(t_1,t_2)=(\varphi_{t_1}^{v_1})_{t_2}^{v_2},$$

等等．这样 $\varphi=\varphi(v_1)=\varphi(v_1,v_2)$．在我们用 "$\varphi(x)$" 的时候，我们不用太担心 x 能不能代入到 φ 中的 v_1．因为如果不能，我们可以让 $\varphi(x)$ 为 $\psi_x^{v_1}$，其中 ψ 是 φ 的某一个字母变换式．

现在假设我们有下面的一般情况：

L_0 是语言．(语言是一些有实际用处的参数集合，通常可以用等号来讨论．)

T_1 是 (可能不同) 语言 L_1 中的理论，其中包含等号．

定义　L_0 在 T_1 中的**解释** π 是 L_0 的参数集上的一个函数，满足

(1) π 把 L_1 中的一个公式 $\pi_\forall$ 指派给 $\forall$，$\pi_\forall$ 中至多只有 v_1 是自由出现的，使得

(i)
$$T_1\models\exists v_1\pi_\forall.$$

(其中的思想是，在 T_1 的任意模型中，公式 $\pi_\forall$ 定义了一个非空集合作为 L_0 结构的论域．)

(2) π 为每个 n 元谓词参量 P 指派了 L_1 中的一个公式 π_P，公式中至多只有 $v_1,\cdots,v_n$ 是自由出现的．

167

(3) π 为每个 n 元函数符号 f 指派了 L_1 中的一个公式 π_f，公式中至多只有 $v_1,\cdots,v_n,v_{n+1}$ 是自由出现的，使得

(ii)
$$\begin{aligned}&T_1\models\forall v_1\cdots\forall v_n(\pi_\forall(v_1)\to\cdots\to\pi_\forall(v_n)\to\\ &\qquad\exists x(\pi_\forall(x)\wedge\forall v_{n+1}(\pi_f(v_1,\cdots,v_{n+1})\leftrightarrow v_{n+1}=x))).\end{aligned}$$

(用汉语表述这个公式为，"对于 $\pi_\forall$ 定义的集合中的所有 $\bar{v}$，存在一个唯一的 x 使得 $\pi_f(\bar{v},x)$ 以及 x 都在 $\pi_\forall$ 定义的集合中．" 这个句子保证了在 T_1 的任何模型中，π_f 在 $\pi_\forall$ 定义的论域上是一个函数．对于常量符号 c，有 $n=0$ 且 (ii) 变为

$$T_1\models\exists x(\pi_\forall(x)\wedge\forall v_1(\pi_c(v_1)\leftrightarrow v_1=x)).$$

换句话说，π_c 定义了一个单点，它也在 $\pi_\forall$ 定义的集合中．)

例如，如果 L_0 是 $(\mathbb{N};0,S)$ 的语言，T_1 是 $(\mathbb{Z};+,\cdot)$ 的理论，那么我们有

$$\pi_\forall(x)=\exists y_1\exists y_2\exists y_3\exists y_4 x = y_1\cdot y_1 + y_2\cdot y_2 + y_3\cdot y_3 + y_4\cdot y_4,$$
$$\pi_{\mathbf{0}}(x)=x+x=x,$$
$$\pi_{\mathbf{S}}(x,y)=\forall z(z\cdot z=z\wedge z+z\neq z\to x+z=y).$$

(这里我们使用了一个事实：在 $(\mathbb{Z};+,\cdot)$ 中可以有效地定义结构 $(\mathbb{N};0,S)$.)

如果 L_0 和 L_1 相等，那么 π 就是平凡的恒等解释，其中

$$\pi_\forall = v_1 = v_1,$$

$$\pi_P = Pv_1\cdots v_n,$$

$$\pi_f = fv_1\cdots v_n = v_{n+1}.$$

无论 T_1 是什么理论，条件 (i) 和 (ii) 都能满足.

现在假设 π 是一个解释，$\mathfrak{B}$ 是 T_1 的模型. 对于 L_0 有一种自然的方法从 $\mathfrak{B}$ 中"提取"结构 ${}^\pi\mathfrak{B}$. 即，令

$|{}^\pi\mathfrak{B}| = \pi_\forall$ 在 $\mathfrak{B}$ 中定义的集合，

$P^{\pi\mathfrak{B}} = \pi_P$ 在中定义的关系，即 π_P 在 $|\pi^{\mathfrak{B}}|$ 上的限制，

$f^{\pi\mathfrak{B}}(a_1,\cdots,a_n) =$ 唯一的 b 满足 $\models_{\mathfrak{B}} \pi_f[\![a_1,\cdots,a_n,b]\!]$，其中 $a_1,\cdots,a_n$ 在 $|{}^\pi\mathfrak{B}|$ 中.

根据解释定义中的条件 (i)，$|{}^\pi\mathfrak{B}| \neq \varnothing$. 并且根据条件 (ii)，$f^{\pi\mathfrak{B}}$ 的定义要有意义，即，存在唯一一个 b 满足上面的条件. 因此，${}^\pi\mathfrak{B}$ 就是语言 L_0 的结构.

我们用下面的式子来定义 L_0 句子的集合 $\pi^{-1}[T_1]$，

$$\begin{aligned}\pi^{-1}[T_1] &= \mathrm{Th}\{{}^\pi\mathfrak{B} \mid \mathfrak{B} \in \mathrm{Mod}\,T_1\}\\ &= \{\sigma \mid \sigma \text{是在每个 } {}^\pi\mathfrak{B} \text{ 中取值均为真的 } L_0 \text{ 句子,}\\ &\qquad {}^\pi\mathfrak{B} \text{ 可以从 } T_1 \text{ 的模型 } \mathfrak{B} \text{ 中得到}\}.\end{aligned}$$

168

这是一个理论，就像任意类 $\mathcal{K}$ 的理论 $\mathrm{Th}\,\mathcal{K}$ 一样. 它是一个可满足的理论当且仅当 T_1 是可满足的.

例 在本节的较前部分，我们有一个理论 T 包含句子

$$\forall v_1 \exists! v_2 \varphi. \qquad (\varepsilon)$$

所讨论的语言是包含函数符号 f 的较大语言 L^+. f 的"定义"由下面的 L^+ 句子给出：

$$\forall v_1 \forall v_2 (fv_1 = v_2 \leftrightarrow \varphi). \qquad (\delta)$$

我们证明了对于原始语言 T 中的句子 σ，如果 $T;\delta \models \sigma$，那么 $T \models \sigma$.

我们已经有了一个从 L^+ 到 T 的解释 π. 除了 f，π 在其他参量上的解释都是恒等的，公式 π_f 就是 φ. 要证明 π 是一个解释只需要一个事实 $T \models \varepsilon$. 对于 T 的任意模型 $\mathfrak{A}$，${}^\pi\mathfrak{A}$ 就是以前被记为 $(\mathfrak{A}, F)$ 的结构，它是 $T;\delta$ 的模型.

我们断言

$$\pi^{-1}[T] = \mathrm{Cn}(T;\delta).$$

首先我们可以看出 $T;\delta$ 的任意模型 $\mathfrak{B}$ 等于 ${}^\pi\mathfrak{A}$，其中 $\mathfrak{A}$ 是 $\mathfrak{B}$ 在 T 的语言上的限制. 因此，对于一个 L^+ 句子 σ，

$$\begin{aligned}\sigma \in \pi^{-1}[T] &\Leftrightarrow \models_{{}^\pi\mathfrak{A}} \sigma \quad \text{对于 } T \text{ 的每个模型 } \mathfrak{A}\\ &\Leftrightarrow \models_{\mathfrak{B}} \sigma \quad \text{对于 } T;\ \delta \text{ 的每个模型 } \mathfrak{B}\\ &\Leftrightarrow T;\delta \models \sigma.\end{aligned}$$

2.7.3 语法翻译

在前面关于解释的一小节中，我们谈到任意模型等．但读者可能已经注意到，L_0 到 T_1 的解释中还有一些关于更加实际的问题要说明．主要有：对于 L_0 中给定的公式 φ，我们能够找到 L_1 中的一个公式 φ^π，从某种意义上说，这个公式恰好与 φ 相对应．我们可以对 φ 递归地定义 φ^π.

首先，考虑 L_0 中的原子公式 α．例如，如果 α 是

$$Pfgx,$$

那么 α 逻辑等价于

$$\forall y(gx = y \to \forall z(fy = z \to Pz)).$$

并且我们可以认为 α^π 是 L_1 公式

169

$$\forall y(\pi_g(x,y) \to \forall z(\pi_f(y,z) \to \pi_P(z))).$$

一般这样的原子公式 α 从右边做起．对于某个 n 元 g，找到最右边出现的函数符号，一般是 $gx_1 \cdots x_n$ 形式的段．(例子是 $n=1$ 的情况．) 用某个新变元 y 来代替它，然后在前面加上 $\forall y(\pi_g(x_1, \cdots x_n, y) \to$，然后继续找第二处函数符号．最后，把谓词符号 P(如果是一个参数) 用带有合适变元的 π_P(带有正确的变量) 代替．

通过对 α 中出现函数符号的次数进行递归定义，我们可以更清楚地看清 α^π．如果函数符号出现的次数是零，那么 α 是 $Px_1 \cdots x_n$，α^π 是 $\pi_P(x_1 \cdots x_n)$．否则，找到最右边出现的函数符号 g，如果 g 是 n 元符号，那么它是以 $gx_1 \cdots x_n$ 形式开头的段．把这个段用某个新变元 y 来代替，得到的公式称为 $\alpha_y^{gx_1 \cdots x_n}$．那么 α^π 就是

$$\forall y(\pi_g(x_1, \cdots, x_n, y) \to (\alpha_y^{gx_1 \cdots x_n})^\pi).$$

例如，

$$\begin{aligned}(Pfgx)^\pi &= \forall y(\pi_g(x,y) \to (Pfy)^\pi) \\ &= \forall y(\pi_g(x,y) \to \forall z(\pi_f(y,z) \to (Pz)^\pi)) \\ &= \forall y(\pi_g(x,y) \to \forall z(\pi_f(y,z) \to \pi_P(z))).\end{aligned}$$

非原子公式的解释也是用一种自然的方式定义的．$(\neg\,\varphi)^\pi$ 是 $(\neg\,\varphi^\pi)$，$(\varphi \to \varphi)^\pi$ 是 $(\varphi^\pi \to \varphi^\pi)$，而且 $(\forall x\varphi)^\pi$ 是 $\forall x(\pi_\forall(x) \to \varphi^\pi)$．(这样量词就与 $\pi_\forall$"有关"．)

下面的引理详细说明了所谓的 φ^π 和 φ"表达的内容相同"．

引理 27B 设 π 是 L_0 到 T_1 的解释，$\mathfrak{B}$ 是 T_1 的模型．对于 L_0 的任意公式 φ 和由变元到 $|{}^\pi\mathfrak{B}|$ 的任意映射 s，

$$\models_{{}^\pi\mathfrak{B}} \varphi[s] \quad \text{iff} \quad \models_{\mathfrak{B}} \varphi^\pi[s].$$

这并不是一个很难的引理，它只是说明了 φ^π 被正确地定义了．

证明 我们对 φ 作归纳，但只有 φ 是原子公式 α 的情况不是平凡的．对于 α，我们对 α 中出现函数符号的次数进行归纳，次数为零的情况是显然的．如果不为零，

$$\alpha^\pi = \forall y(\pi_g(x,y) \to \beta^\pi),$$

其中 $\beta^y_{gx} = \alpha$. (假设 g 是一元符号，这个符号已经足够坏.) 令

$$\begin{aligned} b &= \text{唯一的 } b \text{ 使得} \models_{\mathfrak{B}} \pi_g[\![s(x), b]\!] \\ &= g^{^\pi\mathfrak{B}}(s(x)). \end{aligned}$$

170

则

$$\begin{aligned} \models_{\mathfrak{B}} \alpha^\pi[s] &\Leftrightarrow \models_{\mathfrak{B}} \beta^\pi[s(y|b)] \\ &\Leftrightarrow \models_{^\pi\mathfrak{B}} \beta[s(y|b)] \quad \text{由演绎假设} \\ &\Leftrightarrow \models_{^\pi\mathfrak{B}} \beta^y_{gx}[s] \quad \text{由替换引理} \\ &\Leftrightarrow \models_{^\pi\mathfrak{B}} \alpha[s]. \end{aligned}$$

■

下面的推论说明了 $\pi^{-1}[T_1]$ 这个符号选取的合理性.

推论 27C 对于 L_0 句子 σ,

$$\sigma \in \pi^{-1}[T_1] \quad \text{iff} \quad \sigma^\pi \in T_1.$$

证明 由定义我们有

$$\begin{aligned} \sigma \in \pi^{-1}[T_1] &\Leftrightarrow \text{对于 } T_1 \text{ 的每个模型} \mathfrak{B}, \models_{^\pi\mathfrak{B}} \sigma \\ &\Leftrightarrow \text{对于 } T_1 \text{ 的每个模型} \mathfrak{B}, \models_{\mathfrak{B}} \sigma^\pi, \text{根据引理 27B} \\ &\Leftrightarrow T_1 \models \sigma^\pi. \end{aligned}$$

■

定义 理论 T_0 到 T_1 的 **解释**π 是 T_0 的语言到 T_1 的解释 π, 满足

$$T_0 \subseteq \pi^{-1}[T_1].$$

换句话说，对于 L_0 句子 σ,

$$\sigma \in T_0 \Rightarrow \sigma^\pi \in T_1.$$

$\pi^{-1}[T_1]$ 是到 T_1 的解释为 π 的理论中最大的一个. 如果 $T_0 = \pi^{-1}[T_1]$, 那么我们有 $\sigma \in T_0 \Leftrightarrow \sigma^\pi \in T_1$.

在这种情况下，我们称 π 为 T_0 到 T_1 的 *忠实* 解释.

回到较早的一个例子, 考虑结构 $(\mathbb{N}; 0, S)$ 和 $(\mathbb{Z}; +, \cdot)$. 我们已经有了一个到理论 $\mathrm{Th}(\mathbb{Z}; +, \cdot)$ 的解释 π, 其中

$$\begin{aligned} \pi_\forall(x) &= \exists y_1 \exists y_2 \exists y_3 \exists y_4 x = y_1 \cdot y_1 + y_2 \cdot y_2 + y_3 \cdot y_3 + y_4 \cdot y_4, \\ \pi_0(x) &= x + x = x, \\ \pi_S(x, y) &= \forall z(z \cdot z = z \wedge z + z \neq z \to x + z = y). \end{aligned}$$

我们断言 π 是理论 $\mathrm{Th}(\mathbb{N}; 0, S)$ 到 $\mathrm{Th}(\mathbb{Z}; +, \cdot)$ 的忠实解释. 在这种情况下，$^\pi(\mathbb{Z}; +, \cdot)$ 就是结构 $(\mathbb{N}; 0, S)$. 因此

$$\models_{(\mathbb{N};0,S)} \sigma \Leftrightarrow \models {}_{\pi(\mathbb{Z};+,\cdot)}\sigma \Leftrightarrow \models_{(\mathbb{Z};+,\cdot)} \sigma^\pi.$$

171 在第 3 章中，我们就能证明不存在从 $\text{Th}(\mathbb{Z};+,\cdot)$ 到 $\text{Th}(\mathbb{N};0,S)$ 的解释. 因此前一个理论要比后一个强.

最后，我们回到本节开始的情况. 假设 T 是包含句子 ε 的理论，其中

$\varepsilon = \forall v_1 \exists! v_2 \varphi$;

$\delta = \forall v_1 \forall v_2 (f v_1 = v_2 \leftrightarrow \varphi)$;

$L^+ =$ 向 T 的语言中添加新的函数符号 f 后得到的语言;

$\pi = L^+$ 到 T 的解释，除了 f 外，它在其他所有的参量上都是恒等解释，$\pi_f = \varphi$.

实际上，π 是 $\text{Cn}(T;\delta)$ 到 T 的忠实解释，这是因为正如前面所提到的，

$$\pi^{-1}[T] = \text{Cn}(T;\delta).$$

我们可以得出一个附加结论：这个定义是*可消去的*.

定理 27D　假设条件如上所述，那么对于任意的 L^+ 句子 σ，我们能够在原始语言中找到句子 σ^π，使得

(a) $T;\delta \vDash (\sigma \leftrightarrow \sigma^\pi)$.

(b) $T;\delta \vDash \sigma \Leftrightarrow T \vDash \sigma^\pi$.

(c) 如果 f 在 σ 中不出现，那么 $\vDash (\sigma \leftrightarrow \sigma^\pi)$.

证明　(c) 可以直接从下列事实得出：π 除了 f 之外的所有参量的恒等解释. (b) 重申了 π 是 $\text{Cn}\,(T;\delta)$ 到 T 的忠实解释这一结果. 由于 π 是忠实的，因此要证明 (a) 只要证明

$$T \vDash (\sigma \leftrightarrow \sigma^\pi)^\pi.$$

这可以从 (c) 得到，因为 $(\sigma \leftrightarrow \sigma^\pi)^\pi$ 就是 $(\sigma^\pi \leftrightarrow \sigma^{\pi\pi})$，二者是等值的. ■

习题

1. 假设 L_0 和 L_1 含有相同参量的语言，但 L_0 中有一个 n 元函数符号 f 不在 L_1 中，同时 L_1 中有一个 $(n+1)$ 元谓词符号 P 不在 L_0 中. 证明对于 L_0 的任意一个理论 T，存在一个从 T 到 L_1 的某个理论的忠实解释.
2. 设 L_0 是含有等号及二元函数符号 $+$ 和 $\cdot$ 的语言，L_1 也一样，只不过用三元谓词符号表示加法和乘法. 令 $\mathfrak{N}_i = (\mathbb{N};+,\cdot)(i=0,1)$ 分别是包含自然数和加法、乘法的语言 L_i 的结构. 证明在 $\mathfrak{N}_0$ 中由 L_0 公式定义的任意关系都能在 $\mathfrak{N}_1$ 中由 L_1 公式定义.

172

3. 证明一个完全理论到一个可满足理论的解释是忠实的.

2.8　非标准分析[1]

在 17 世纪，莱布尼兹和牛顿根据非零的无穷小量提出了微分和积分的思想. 牛顿在他的计算中使用了一个无穷小量 o，它可以和任意一个有限数相乘所得结果仍然是无穷小，但 o 可以作为分母，因此它不是零. 莱布尼兹的 $\mathrm{d}x$ 小于任意一个指定的量，同时也是非零的.

1. 忽略本节并不影响全书的连续性.

这种思想并不容易理解和接受. 无穷小的工作在整个 18 世纪都遭到包括 Bishop Berkeley、D'Alembert 等人的攻击和质疑，但欧拉等人热心地进行了验证. 正是欧拉大胆、自由地把无穷小量引入到数学中，才创立了近代微积分学. 直到 19 世纪，微积分才发展成现在课本中的形式，并且有了严格定义的极限，这样争论才平息下来.

1961 年，Abraham Robinson 提出了一种处理极限的新方法，使得无穷小量有了实质的载体. 这种方法把使用无穷小量的直观益处和现代数学中的严格性标准结合起来. 它的基本思想就是利用实数理论的非标准模型来构造无穷小量.

2.8.1 $^*\mathfrak{R}$ 的构造

在本节中，我们要使用一个很大的一阶语言，语言中除了包含符号 $+, \cdot$ 和 $<$ 外，还要增加幂乘函数和绝对值函数. 我们所讨论的对象是实数集 $\mathbb{R}$，其上的关系和函数符号仍沿用前面的定义. 这样，我们就有了一个包含等号的语言以及下列参量：

(0) $\forall$ 表示“对于所有的实数.”

(1) 对于 $\mathbb{R}$ 上的每个 n 元关系 R 都有一个 n 元谓词符号 $\boldsymbol{P}_R$ 与之对应.

(2) 对于每个 $r \in \mathbb{R}$，存在一个常量符号 $\boldsymbol{c}_r$.

(3) 对于 $\mathbb{R}$ 上的每个 n 元运算 F 都有一个 n 元函数符号 $\boldsymbol{f}_F$ 与之对应. 173

对于这个语言，我们有一个标准结构 $\mathfrak{R}$，满足 $|\mathfrak{R}| = \mathbb{R}$，$\boldsymbol{P}_R^{\mathfrak{R}} = R$，$\boldsymbol{c}_r^{\mathfrak{R}} = r$，$\boldsymbol{f}_F^{\mathfrak{R}} = F$. 但我们现在要利用紧致性定理来构造它的非标准结构. 令 Γ 是下面的集合

$$\text{Th } \mathfrak{R} \cup \{\boldsymbol{c}_r \boldsymbol{P}_< v_1 | r \in \mathbb{R}\}.$$

(此处 $\boldsymbol{c}_r \boldsymbol{P} < v_1$ 是“r 小于 v_1”公式化.) 只要把某个足够大的实数指派给 v_1，Γ 的任意有限子集都能在 $\mathfrak{R}$ 中得到满足. 因此，根据紧致性定理，存在一个结构 $\mathfrak{A}$ 及元素 $a \in |\mathfrak{A}|$，使得如果将 a 指派给 v_1，那么 Γ 在 $\mathfrak{A}$ 中可满足. 由于 $\mathfrak{A}$ 是 Th $\mathfrak{R}$ 的一个模型，因此有 $\mathfrak{A} \equiv \mathfrak{R}$. 也存在从 $\mathfrak{R}$ 到 $\mathfrak{A}$ 内的但不是到上的同构 h，h 由下式定义

$$h(r) = \boldsymbol{c}_r^{\mathfrak{A}}.$$

为了验证这的确是个同构，我们要用到 $\mathfrak{A} \equiv \mathfrak{R}$ 这一事实. h 是一对一的，这是因为对于 $r_1 \neq r_2$，句子 $\boldsymbol{c}_{r_1} \neq \boldsymbol{c}_{r_2}$ 在 $\mathfrak{R}$ 中成立，进而在 $\mathfrak{A}$ 中成立. h 保持二元关系 $R(= \boldsymbol{P}_R^{\mathfrak{R}})$，因为对于 $\mathbb{R}$ 中任意的 r 和 s，

$$\begin{aligned}
\langle r, s\rangle \in \boldsymbol{P}_R^{\mathfrak{R}} &\Leftrightarrow \models_{\mathfrak{R}} \boldsymbol{P}_R \boldsymbol{c}_r \boldsymbol{c}_s \\
&\Leftrightarrow \models_{\mathfrak{A}} \boldsymbol{P}_R \boldsymbol{c}_r \boldsymbol{c}_s \\
&\Leftrightarrow \langle \boldsymbol{c}_r^{\mathfrak{A}}, \boldsymbol{c}_s^{\mathfrak{A}}\rangle \in \boldsymbol{P}_R^{\mathfrak{A}} \\
&\Leftrightarrow \langle h(r), h(s)\rangle \in \boldsymbol{P}_R^{\mathfrak{A}}.
\end{aligned}$$

对于 n 元关系也有类似的讨论. 下面我们要证明 h 保持任意函数 $F(= \boldsymbol{f}_F^{\mathfrak{R}})$. 为了简便，我们假设 F 是二元运算. 考虑 $\mathbb{R}$ 中任意的 r 和 s，令 $t = F(r, s)$. 那么，

$$\begin{aligned}
h(\boldsymbol{f}_F^{\mathfrak{R}}(r, s)) &= h(F(r, s)) \\
&= h(t) \\
&= \boldsymbol{c}_t^{\mathfrak{A}}.
\end{aligned}$$

这样句子 $c_t = f_F c_r c_s$ 在 $\mathfrak{R}$ 中成立，进而在 $\mathfrak{A}$ 中成立. 这样

$$\begin{aligned} c_t^{\mathfrak{A}} &= f_F^{\mathfrak{A}}(c_r^{\mathfrak{A}}, c_s^{\mathfrak{A}}) \\ &= f_F^{\mathfrak{A}}(h(r), h(s)). \end{aligned}$$

因此 h 保持 f_F. 对于常量符号，根据 h 的定义，

$$\begin{aligned} h(c_r^{\mathfrak{R}}) &= h(r) \\ &= c_r^{\mathfrak{A}}. \end{aligned}$$

由于 $\mathfrak{R}$ 已经被同构映射到 $\mathfrak{A}$ 中，所以我们能够找到另一个与 $\mathfrak{R}$ 同构的结构 $^*\mathfrak{R}$，使得 $\mathfrak{R}$ 是 $^*\mathfrak{R}$ 的子结构. 这个想法只是用元素 r 代替 $\mathfrak{A}$ 中的元素 $c_r^{\mathfrak{A}}$ (我们可以假设 $|\mathfrak{A}| \cap \mathbb{R} = \varnothing$). 细节之处见 2.2 节的习题 24. 由于 $^*\mathfrak{R}$ 同构于 $\mathfrak{A}$，所以存在一个元素 $b \in |^*\mathfrak{R}|$，使得当把 b 指派给 v_1 时，$^*\mathfrak{R}$ 满足 Γ. 特别地，$^*\mathfrak{R} \equiv \mathfrak{R}$.

174

我们进一步简化繁琐的记号，用星号表示 $\mathfrak{R}$ 到 $^*\mathfrak{R}$ 的嵌入.

(1) 对于 $\mathbb{R}$ 上的 n 元关系 R，用 $^*R$ 表示在 $^*\mathfrak{R}$ 中指派给符号 P_R 的关系 $P_R^{^*\mathfrak{R}}$. 特别地，$\mathbb{R}$ 是 $\mathbb{R}$ 上的一元关系. 它的像 $^*\mathbb{R}$ 等于 $^*\mathfrak{R}$ 的论域，这是因为句子 $\forall x P_{\mathbb{R}} x$ 在 $\mathfrak{R}$ 中取值为真，进而在 $^*\mathfrak{R}$ 中的取值为真. 由于 $\mathfrak{R}$ 是 $^*\mathfrak{R}$ 的子结构，因此我们有，每个关系 R 等于 $^*\mathfrak{R}$ 在 $\mathbb{R}$ 上的限制.

(2) 对于 $\mathbb{R}$ 上的每个 n 元运算 F，用 $^*F$ 表示在 $^*\mathfrak{R}$ 中指派给符号 f_F 的运算 $f_R^{^*\mathfrak{R}}$. F 就是 $^*F$ 在 $\mathbb{R}$ 上的限制.

由于 $c_r^{^*\mathfrak{R}} = r$，所以对此我们不需要特殊的记号.

我们用一种一般的方法来证明一个关系 $^*R$ 或运算 $^*F$ 具有的性质，这种方法在本节的剩余部分经常用到. 不难发现 (1)R 或 F 具有某种性质，(2) 这种性质可以用语言中的一个句子来表述，并且 (3)$\mathfrak{R} \equiv {}^*\mathfrak{R}$.

例如，二元关系 $^*<$ 在 $^*\mathbb{R}$ 中是传递的. 这是因为 $<$ 是传递的，并且这个性质可以用下面的句子表达：

$$\forall x \forall y \forall z (x P_< y \to y P_< z \to x P_< z).$$

根据类似的理由，$^*<$ 满足 $^*\mathbb{R}$ 上的三分法，这样它就是 $^*\mathbb{R}$ 上的一种序关系.

作为另一个例子，我们可以证明二元关系 $^*+$ 在 $^*\mathbb{R}$ 上是可交换的. 因为 $+$ 是可交换的并且交换律可以用句子表示出来. 把这个理由用到域的每条公理上，我们就得到 $(^*\mathbb{R}; 0, 1, ^*+, ^*\cdot)$ 是一个域.

由于经常用到这个方法，我们把它看作是理所当然的. 例如，如果假设对于 $^*\mathbb{R}$ 中的 a 和 b，有 $^*|a \,^*\!+ b| \,^*\!\leqslant \,^*|a| \,^*\!+ \,^*|b|$，读者应该把它看作是由这个方法得到的.

我们可以得到 $\mathbb{R} \subseteq {}^*\mathbb{R}$，但 $\mathbb{R} \neq {}^*\mathbb{R}$. 因为我们有某个元素 b 满足 $\vDash_{^*\mathfrak{R}} c_r P_< v_1 [\![b]\!]$；即 $r \,^*\!< b$. 这样 b 是无穷大的，比任意一个标准的 r，即任意的 $r \in \mathbb{R}$，都要大 (在序 $^*<$ 之下). 它的倒数 $1 ^*\!/ b$ 就是一个无穷小量.

那些不能在 $\mathfrak{R}$ 的语言中表达的性质可能也不能在 $^*\mathfrak{R}$ 中表达. 上确界就是这样的性质. 存在 $^*\mathbb{R}$ 的非空有界子集 S 没有上确界 (在 $^*<$ 序下). 例如，$\mathbb{R}$ 就是这样一个 $^*\mathbb{R}$ 的子集. 上一段中无限大的 b 是它的上界，但它却没有上确界；见习题 7.

我们用下面的式子来定义 有限 元的集合 $\mathcal{F}$:

$$\mathcal{F}=\{x\in {}^*\mathbb{R}|\ {}^*|x|{}^*< y \quad \text{对某个 } y\in\mathbb{R}\}.$$

类似地，可以用下面的公式来定义 无穷小量 的集合 $\mathcal{I}$: 175

$$\mathcal{I}=\{x\in {}^*\mathbb{R}|\ {}^*|x|{}^*< y \quad \text{对于所有正数 } y\in\mathbb{R}\}.$$

如果 $A\subseteq\mathbb{R}$ 是无界的，那么 $^*A$ 包含了无穷大的元素. 因为句子“对于任意实数 r，存在一个元素 $a\in A$ 比 r 大”的取值为真且可以形式化. 取一个无穷大的正数 b，$^*A$ 中一定存在一个更大的元素. 例如，$^*\mathbb{N}$ 中就包含有无穷大的数.

而唯一一个标准的无穷小量是 0，即 $\mathbb{R}\cap\mathcal{I}$ 的唯一元素. 但非标准的无穷小量还有许多，因为任意一个无穷大数的倒数都是无穷小的.

2.8.2 代数性质

我们在下一个定理中列出了 $\mathcal{F}$ 和 $\mathcal{I}$ 的一些常用的代数性质.

定理 28A (a) $\mathcal{F}$ 在加法 $^*+$，减法 $^*-$ 和乘法 $^*\cdot$ 下是封闭的.
(b) $\mathcal{I}$ 在加法 $^*+$，减法 $^*-$ 和 $\mathcal{F}$ 上的数乘下是封闭的:

$$x\in\mathcal{I}\text{并且}z\in F\Rightarrow x\,{}^*\!\cdot z\in\mathcal{I}.$$

在代数中，(a) 表示 $\mathcal{F}$ 是域 $^*\mathbb{R}$ 的子环，(b) 表示 $\mathcal{I}$ 是环 $\mathcal{F}$ 的理想. 后面我们将看到什么是商环 $\mathcal{F}/\mathcal{I}$.

证明 (a) 令 x 和 y 是有限的，因此存在 $\mathbb{R}$ 中的标准数 a 和 b，使得 $^*|x|{}^*< a$，$^*|y|{}^*< b$，那么

$$^*|x\,{}^*\!\pm y|{}^*\leqslant {}^*|x|{}^*\!+{}^*|y|{}^*< a+b\in\mathbb{R},$$

这里 $x\,{}^*\!+y$，$x\,{}^*\!-y$ 都是有限的. 同时,

$$^*|x\,{}^*\!\cdot y|{}^*< a\cdot b\in\mathbb{R},$$

其中 $x\,{}^*\!\cdot y$ 也是有限的.

(b) 令 x 和 y 是无穷小量，那么对于任意的标准正数 a，都有 $^*|x|{}^*< a/2$ 并且 $^*|y|{}^*< a/2$. 因此

$$^*|x\,{}^*\!\pm y|{}^*< a/2+a/2=a,$$

使得 $x\,{}^*\!+y$ 和 $x\,{}^*\!-y$ 是无穷小量. 如果 z 是有限的，那么对某个标准 b，有 $^*|z|{}^*< b$. 由于 x 是无穷小量，我们有 $^*|x|{}^*< a/b$，因此 $^*|x\,{}^*\!\cdot z|{}^*< (a/b)b=a$. 这样，$x\,{}^*\!\cdot z$ 也是无穷小量. ■ 176

定义 x无限趋近y(记作 $x\simeq y$) 当且仅当 $x\,{}^*\!-y$ 是无穷小量.

定理 28B (a) $\simeq$ 是 $^*\mathbb{R}$ 上的等价关系.
(b) 如果 $u\simeq v$ 并且 $x\simeq y$，那么 $u\,{}^*\!+x\simeq v\,{}^*\!+y$ 且 $^*\!-u\simeq {}^*\!-v$.
(c) 如果 $u\simeq v$，$x\simeq y$ 并且 x,y,u,v 都是有限的，那么 $u\,{}^*\!\cdot x\simeq v\,{}^*\!\cdot y$.

证明 这是上个定理中 (b) 的结果 ($\mathcal{I}$ 是 $\mathcal{F}$ 中的一个理想).

(a) $\simeq$ 是自反的，因为 0 是无穷小量. $\simeq$ 是对称的，因为一个无穷小量的相反数 ($^*-$) 还是无穷小量. 最后，假设 $x \simeq y$, $y \simeq z$, 那么

$$x\,^*\!-z = (x\,^*\!-y)\,^*\!+(y\,^*\!-z) \in \mathcal{I},$$

因为 $\mathcal{I}$ 在加法下封闭.

(b) 如果 $u \simeq v$ 并且 $x \simeq y$, 那么

$$(u\,^*\!+x)\,^*\!-(v\,^*\!+y) = (u\,^*\!-v)\,^*\!+(x\,^*\!-y) \in \mathcal{I},$$

因为 $\mathcal{I}$ 在加法下封闭. 同样 $^*-u \simeq {}^*-v$, 因为 $\mathcal{I}$ 在否定下封闭.

(c)
$$\begin{aligned}(u\,^*\!\cdot x)\,^*\!-(v\,^*\!\cdot y) &= (u\,^*\!\cdot x)\,^*\!-(u\,^*\!\cdot y)\,^*\!+(u\,^*\!\cdot y)\,^*\!-(v\,^*\!\cdot y)\\ &= u\,^*\!\cdot(x\,^*\!-y)\,^*\!+(u\,^*\!-v)\,^*\!\cdot y \in \mathcal{I}\end{aligned}$$

因为 $\mathcal{I}$ 在 $\mathcal{F}$ 上的数乘是封闭的. ■

对于标准数 r 和 s, 我们有 $r \simeq s$ 当且仅当 $r = s$, 这是因为 0 是唯一一个标准无穷小量.

引理 28C 如果 $x \not\simeq y$ 并且二者中至少有一个是有限的，那么一定存在一个标准数 q 严格位于 x 和 y 之间.

证明 不妨假设 $x\,^*\!< y$. 实际上我们可以进一步假设 $0\,^*\!< x\,^*\!< y$. $x\,^*\!< y\,^*\!\leqslant 0$ 的情况是类似的, $x\,^*\!< 0\,^*\!< y$ 的情况是显然的. 由于 $x \not\simeq y$, 因此存在一个标准数 b 使得 $0 < b\,^*\!< y\,^*\!-x$. 又因为 x 是有限的，因此存在某个正整数 m 使得 $x\,^*\!< mb$, 我们取满足条件的元素中最小的一个为 m, 那么 $x\,^*\!< mb\,^*\!< y$. (根据 m 的最小性, $(m-1)b\,^*\!\leqslant x$, 因此 $mb\,^*\!\leqslant x\,^*\!+b\,^*\!< y$.)■

定理 28D 每个 $x \in \mathcal{F}$ 都无限趋近于唯一一个 $r \in \mathfrak{R}$.

证明 对于每个 $x \in \mathcal{F}$, 小于 x 的标准数集合

$$S = \{y \in \mathbb{R} | y\,^*\!< x\}$$

177 在 $\mathbb{R}$ 中显然有上界. 令 r 是它的上确界，我们断言 $x \simeq r$.

如果 $x \not\simeq r$, 那么根据引理存在一个位于 x 和 r 之间的标准数 q. 如果 $r < q\,^*\!< x$, 那么 r 不可能是 S 的上界. 如果 $x\,^*\!< q < r$, 那么 q 也是 S 的上界，这与 r 的最小性相矛盾. 因此 $x \simeq r$.

上面证明了 r 的存在性. 至于唯一性，我们注意到，如果 $x \simeq r$ 并且 $x \simeq s$, 那么 $r \simeq s$. 对于标准数 x 和 s, 这就意味着 $r = s$. ■

推理 28E 每个有限数 x 都可以唯一分解成 $x = s\,^*\!+i$ 的形式，其中 s 是一个标准数, i 是一个无穷小量.

我们称 s 为 x 的 **标准部分**, 记作 st(x) (x 的标准部分有时也记作 $°x$). 当然对于标准数 r, st $(r) = r$. 下面的定理总结了 st 函数的一些性质.

定理 28F (a) st 把 $\mathcal{F}$ 映射到 $\mathbb{R}$ 上.

(b) $\mathrm{st}(x) = 0$ 当且仅当 x 是无穷小量.

(c) $\mathrm{st}(x \,{}^{*}\!+ y) = \mathrm{st}(x) + \mathrm{st}(y)$.

(d) $\mathrm{st}(x \,{}^{*}\!\cdot y) = \mathrm{st}(x) \cdot \mathrm{st}(y)$.

证明 (a) 和 (b) 是显然的. 由于 $\mathrm{st}(x) \simeq x$ 和 $\mathrm{st}(y) \simeq y$, 根据定理 28B 的 (b), 我们有 $\mathrm{st}(x)+\mathrm{st}(y) \simeq x \,{}^{*}\!+ y$, 因此左边等于 $\mathrm{st}(x \,{}^{*}\!+ y)$. (d) 的证明也是类似的, 只需要利用定理 28B 中的结论 (c). ■

(在代数中, 这个定理断言了 st 是从环 $\mathcal{F}$ 到域 $\mathbb{R}$ 上的同态, 核为 $\mathcal{I}$. 进而, 商环 $\mathcal{F}/\mathcal{I}$ 与实数域 $\mathbb{R}$ 同构.)

本节中后面关于算术运算的符号 ${}^{*}+$、${}^{*}-$、${}^{*}\cdot$ 和 ${}^{*}/$ 一律省略星号.

2.8.3 收敛性

在微积分中收敛性通常是用 ε, δ 语言来描述变量与某个值的接近程度. 在这里我们要给出收敛性的另一种定义, 它描述的是变量无限趋近于极限值.

定义 设 $F: \mathbb{R} \to \mathbb{R}$, 那么 *$F$ 在 a 点收敛于 b* 当且仅当如果 x 无限趋近于 (但不等于)a, 那么 ${}^{*}F(x)$ 无限趋近于 b.

这种定义与一般定义的等价性证明 首先假设 F 在一般意义下在 a 点收敛于 b, 也就是, 对于任意的 $\varepsilon > 0$, 存在一个 $\delta > 0$ 使得对于任意的 x,

$$0 \neq |x - a| < \delta \Rightarrow |b - F(x)| < \varepsilon$$

上述的句子 (关于标准数 ε 和 δ) 可以公式化并且在 ${}^{*}\mathfrak{R}$ 中成立. 既然 x 在 ${}^{*}\mathbb{R}$ 中无限趋近 (但不等于)a, 那么自然有 $0 \neq {}^{*}|x - a|^{*}\!< \delta$. 因此 ${}^{*}|b - {}^{*}F(x)|^{*}\!< \varepsilon$. 由于 ε 是任意的, 所以 $b \simeq {}^{*}F(x)$. [178]

反过来, 假设定义中的条件都满足, 那么对于任意标准数 $\varepsilon > 0$, 句子

$$\text{存在 } \delta > 0 \text{ 使得对于所有 } x,\ 0 \neq |a - x| < \delta \Rightarrow |b - F(x)| < \varepsilon$$

(公式化后) 在 ${}^{*}\mathfrak{R}$ 中成立, 这是因为我们可以取 δ 为无穷小量. 因此这个句子在 $\mathfrak{R}$ 中也成立. ■

说明 1 F 完全有可能在 a 点不收敛于任何数. 另一方面, F 在 a 点至多收敛于一个点 b, 这是因为如果 i 是一个非零无穷小量, 那么 $b = \mathrm{st}({}^{*}F(a + i))$. 我们传统上把这个 b 记作 "$\lim_{x\to a} F(x)$. " 这样

$$\lim_{x\to a} F(x) = \mathrm{st}({}^{*}F(a + i)).$$

说明 2 我们不必要求 $\mathrm{dom}\, F = \mathbb{R}$, a 只要是 $\mathrm{dom}\, F$ 的聚点就够了. (a 是 S 的聚点当且仅当 a 无限趋近但不等于 ${}^{*}S$ 中的某个元素.)

推论 28G F 在 a 点连续当且仅当如果 $x \simeq a$, 那么 ${}^{*}F(x) \simeq F(a)$.

现在我们考虑函数 $F: \mathbb{R} \to \mathbb{R}$ 及标准点 $a \in \mathbb{R}$, 那么微分 $F'(a)$ 是

$$\lim_{h\to 0} \frac{F(a + h) - F(a)}{h}$$

根据极限的定义，微分也可以定义为：$F'(a)=b$ 当且仅当对于任意非零的无穷小量 $\mathrm{d}x$，我们有 $\mathrm{d}F/\mathrm{d}x\simeq b$，其中 $\mathrm{d}F={}^*F(a+\mathrm{d}x)-F(a)$．因此，如果这样的 b 存在 (即 $F'(a)$ 存在)，那么对于任意非零的无穷小量 $\mathrm{d}x$，$F'(a)=\mathrm{st}(\mathrm{d}F/\mathrm{d}x)$．这里 $\mathrm{d}F/\mathrm{d}x$ 是 $\mathrm{d}F$除以$\mathrm{d}x$ 的商．只要利用除法这个事实就大大方便了计算．

例　令 $F(x)=x^2$，那么 $F'(a)=2a$，这是因为

$$\frac{\mathrm{d}F}{\mathrm{d}x}=\frac{(a+\mathrm{d}x)^2-a^2}{\mathrm{d}x}=\frac{2a(\mathrm{d}x)+(\mathrm{d}x)^2}{\mathrm{d}x}=2a+\mathrm{d}x\simeq 2a.$$

定理 28H　如果 $F'(a)$ 存在，那么 F 在 a 点连续．

证明　对于任意非零的无穷小量 $\mathrm{d}x$，我们有

179

$$\frac{{}^*F(a+\mathrm{d}x)-F(a)}{\mathrm{d}x}\simeq F'(a).$$

式子的右边是标准数，因此左边至少是有限的．进而，当左边乘以无穷小量 $\mathrm{d}x$ 时，我们有 ${}^*F(a+\mathrm{d}x)-F(a)\in\mathcal{I}$，即 ${}^*F(a+\mathrm{d}x)\simeq F(a)$． ■

大家应该注意，这个结果并不是经典情况下定理的非标准形式，也不是经典定理的推广．它本身就是经典的定理，只是证明是非标准的，下一个定理的证明也一样．设 $F\circ G$ 为一个函数，它在 a 点的值是 $F(G(a))$．

链式法则　假设 $G'(a)$ 和 $F'(G(a))$ 存在，那么 $(F\circ G)'(a)$ 存在，且等于 $F'(G(a))\cdot G'(a)$．

证明　首先，由于句子 $\forall v_1 \boldsymbol{f}_{F\circ G}v_1=\boldsymbol{f}_F\boldsymbol{f}_G v_1$ 在结构中成立，所以 ${}^*(F\circ G)={}^*F\circ{}^*G$．现在我们考虑任意非零的无穷小量 $\mathrm{d}x$，令

$$\begin{aligned}\mathrm{d}G&={}^*G(a+\mathrm{d}x)-G(a),\\ \mathrm{d}F&={}^*(F\circ G)(a+\mathrm{d}x)-(F\circ G)(a)\\ &={}^*F({}^*G(a+\mathrm{d}x))-F(G(a))\\ &={}^*F(G(a)+\mathrm{d}G)-F(G(a)).\end{aligned}$$

由于 G 在 a 点连续，所以 $\mathrm{d}G\simeq 0$．如果 $\mathrm{d}G\neq 0$，那么由上面的最后一个等式可得 $dF/dG\simeq F'(G(a))$，因此

$$\frac{\mathrm{d}F}{\mathrm{d}x}=\frac{\mathrm{d}F}{\mathrm{d}G}\cdot\frac{\mathrm{d}G}{\mathrm{d}x}\simeq F'(G(a))\cdot G'(a).$$

如果 $\mathrm{d}G=0$，那么 $\mathrm{d}F=0$ 并且 $G'(a)\simeq\mathrm{d}G/\mathrm{d}x=0$，因此我们还有

$$\frac{\mathrm{d}F}{\mathrm{d}x}\simeq F'(G(a))\cdot G'(a).$$ ■

这些定理都是收敛性在无限逼近方面运用的例子，实际上这种方法并不局限于在初等问题中使用．我们可以构造 δ 函数，使它满足 $\int_{-\infty}^{\infty}\delta=1$，但对于 $x\neq 0$，$\delta(x)\simeq 0$．(希尔伯特空间理论中) 数学分析中的原始结果都能用非标准分析的方法得到．随着越来越多的分析学家对这种方法的熟悉，非标准分析的这种方法在将来可能被广泛地使用．

习题

1. ($\mathbb{Q}$ 在 $\mathbb{R}$ 中是稠密的.) 设 $\mathbb{Q}$ 为有理数集合，证明：$^*\mathbb{R}$ 中的每个元素都无限趋近于 $^*\mathbb{Q}$ 中的某个元素.
2. (a) 设 $A \subseteq \mathbb{R}$ 且 $F : A \to \mathbb{R}$. 那么 F 也可看作 $\mathbb{R}$ 上的一个二元关系，证明：$^*F :^* A \to^* \mathbb{R}$.
   (b) 设 $S : \mathbb{N} \to \mathbb{R}$. 我们称 S 收敛于 b 当且仅当对于每个 $\varepsilon > 0$, 存在某个 k 使得对于所有的 $n > k$, 都有 $|S(n) - b| < \varepsilon$. 证明：这个定义等价于下面的条件：对于每个无限大的 $x \in {}^*\mathbb{N}$, $^*S(x) \simeq b$. 180
   (c) 假设 $S_i : \mathbb{N} \to \mathbb{R}$ 并且对于 $i = 1, 2$, S_i 收敛于 b_i. 证明 $S_1 + S_2$ 收敛于 $b_1 + b_2$, 且 $S_1 \cdot S_2$ 收敛于 $b_1 \cdot b_2$.
3. 设 $F : A \to \mathbb{R}$ 是一对一的，其中 $A \subseteq \mathbb{R}$. 证明：如果 $x \in^* A$ 但 $x \notin A$, 那么 $^*F(x) \notin \mathbb{R}$.
4. 设 $A \subseteq \mathbb{R}$. 证明：$A =^* A$ 当且仅当 A 是有限的.
5. (波尔查诺－魏尔斯特拉斯定理) 设 $A \subseteq \mathbb{R}$ 是有界无限集. 证明：存在点 $p \in \mathbb{R}$, p 无限趋近但不等于 $^*A$ 中的某个元素. 提示：令 $S : \mathbb{N} \to A$ 是一对一的，对于无限大的 $x \in^* \mathbb{N}$, 考虑 $^*S(x)$.
6. (a) 证明：$^*\mathbb{Q}$ 的基数至少为 $2^{\aleph_0}$, 其中 $\mathbb{Q}$ 是有理数集合. 提示：利用习题 1.
   (b) 证明：$^*\mathbb{N}$ 的基数至少为 $2^{\aleph_0}$.
7. 设 A 是 $\mathbb{R}$ 的子集，并且没有最大元，那么作为 $^*\mathbb{R}$ 的一个子集，A(根据序 $^*<$) 在 $^*\mathbb{R}$ 中有上界. 证明：A 没有上确界. 181

第3章 不可判定性

3.0 数论

在本章中，我们将关注一种特殊的语言——数论语言．它是包含“等于”和下列参数的一阶语言．

$\forall$，表示“对于任意的自然数”．(自然数集合 $\mathbb{N}$ 为 $\{0,1,2,\cdots\}$，0 是自然数．)

$\mathbf{0}$，常数符号，表示自然数 0．

$\mathbf{S}$，一元函数符号，表示后继函数 $S:\mathbb{N}\to\mathbb{N}$，即，函数 $S(n)=n+1$．

$<$，二元谓词符号，表示 $\mathbb{N}$ 中常用的 (严格) 序关系．

$+,\cdot,\mathbf{E}$，二元函数符号，分别表示加法运算 $+$，乘法运算 $\cdot$，幂乘运算 E．

我们用 $\mathfrak{N}$ 表示该语言的结构，这样，数论结构就可以写成

$$\mathfrak{N}=(\mathbb{N};0,S,<,+,\cdot,E).$$

其中，$|\mathfrak{N}|=\mathbb{N}$，$\mathbf{0}^{\mathfrak{N}}=0$，等等．

182

这个模型的理论，即 **数论**，写作 Th $\mathfrak{N}$．在 3.1 节和 3.2 节的习题中，将研究 $\mathfrak{N}$ 的某些归约模型，即 $\mathfrak{N}$ 在某些子语言上的限制：

$$\mathfrak{N}_S=(\mathbb{N};0,S),$$

$$\mathfrak{N}_L=(\mathbb{N};0,S,<),$$

$$\mathfrak{N}_A=(\mathbb{N};0,S,<,+),$$

最后，将在 3.8 节研究

$$\mathfrak{N}_M=(\mathbb{N};0,S,<,+,\cdot).$$

对于上述的每一个结构，我们都将提出以下几个相同的问题：

(a) 这个结构的理论是可判定的吗？如果是，该理论的公理集是什么样的？这个公理集是否是可数的？

(b) 在这个结构中，$\mathbb{N}$ 的哪些子集是可判定的？

(c) 结构理论的非标准模型是什么样的？(所谓“非标准”是指“与所讨论的结构不同构的”)

我们选择数论 (而不选择其他理论，比如群论) 进行学习是因为：我们能证明数论的某个子理论是一个不可判定的句子集．我们也能得到，任何一个可满足的理论如果包含数论 (例

如，完全数论或者集合论) 的这个子理论，那么这个理论一定是不可判定的．特别地，这样的一个理论不可能既是完全的又是可公理化的．

为了证明我们所找到的数论的子理论是不可判定的，我们还将证明这个子理论足以表示数字序列、数字序列与数之间的编码运算以及判定过程等．在最后一部分内容中，我们将利用对角线法证明我们所选择的子理论是不可判定的．

当然，我们也可以不用数论的子理论作为例子，而改用其他的理论 (例如：有限集理论的子理论)．在这些理论中，我们同样可以方便地讨论判定过程．

在给出数论语言的表达式例子之前，我们先介绍一些符号的习惯用法．为了和目前的用法一致，我们分别用符号

$$x < y, x + y,\ x \cdot y,\ x\mathbf{E}y$$

表示

$$< xy,\ +xy,\ \cdot xy,\ \mathbf{E}xy.$$

对于每个自然数 k，$\mathbf{S}^k\mathbf{0}$(对于 k，是一个数字) 表示一个项：

$$\mathbf{S}^0\mathbf{0} = \mathbf{0},\ \mathbf{S}^1\mathbf{0} = \mathbf{S0},\ \mathbf{S}^2\mathbf{0} = \mathbf{SS0}, \cdots$$

(自然数集合可以由 $\{\mathbf{0}\}$ 和前面介绍的后继函数 $\mathbf{S}$ 生成．) 我们可以用语言描述出每一个自然数，这个性质将是非常有用的． 183

尽管在 $\mathfrak{N}$ 中有可数多个 $\mathbb{N}$ 上的关系，但是几乎所有我们所熟悉的关系都是可定义的．比如：素数集合在 $\mathfrak{N}$ 中可以用以下的公式来定义：

$$v_1 \neq \mathbf{S}^1\mathbf{0} \wedge \forall v_2 \forall v_3 (v_1 = v_2 \cdot v_3 \rightarrow v_2 = \mathbf{S}^1\mathbf{0} \vee v_3 = \mathbf{S}^1\mathbf{0}).$$

以后，我们将会证明许多特殊的关系在 $\mathfrak{N}$ 中都是可以定义的，这是一个很重要的性质．

很自然地，我们希望当语言中的某些参数被忽略时，语言的描述能力仍然能够被严格地保留下来．但有的时候，情况并不如我们所希望的那样．比如，素数集合在 $\mathfrak{N}_A$ 中是不可定义的．另一方面，在 3.8 节中我们将看到，任意一个在 $\mathfrak{N}$ 中可定义的关系在 $\mathfrak{N}_M$ 中也是可定义的．

预览

本章的主要定理是由哥德尔、塔斯基和丘奇提出的，我们将在 3.5 节中给出证明．但在此，我们可以先了解一下这些定理中的思想．首先，我们要比较两个概念：**取值真** 和 **证明**，也就是要区分在 $\mathfrak{N}$ 中取值为真的句子集和可用部分公理加以证明的句子集．

我们给数论语言中的每一个公式 α 指派一个整数 $\sharp\alpha$，此整数称为 α 的哥德尔数．实际上，任何一种能够把不同整数分配给不同公式的方法都是符合我们要求的，在 3.4 节中，我们采用了一种特殊的分配方法．对于我们来说，重要的是能够从公式 α 能行地找到对应的整数 $\sharp\alpha$；反之，也能从 $\sharp\alpha$ 能行地找到公式 α．类似地，对于公式的有限序列 D(例如推理)，我们同样能够给出一个整数 $\mathcal{G}(D)$．注意：对于公式的 **集合** A，我们同样有对应的整数集合 $\{\sharp\alpha | \alpha \in A\}$．

我们将要分别学习 3 种方法：**自代入法**，**对角线法** 和 **可计算性方法**．然而，这 3 种方法的关系要比它们看起来的更加紧密——它们是同一种方法的 3 个方面．

首先，在自代入法中，我们将构造一个句子 σ，可以把它想像成："我是无法证明的．"更特别地，我们有以下结果：

定理 30A　设 $A \subseteq \text{Th}\,\mathfrak{N}$ 是 $\mathfrak{N}$ 中取值为真的句子集，且 A 的哥德尔数集合 $\{\#\alpha|\alpha \in A\}$ 是一个在 $\mathfrak{N}$ 中可定义的集合．则我们可以找到一个在 $\mathfrak{N}$ 中为真的句子 σ，但 σ 不能由 A 推
184
出．

证明　我们来构造一个句子 σ 来表示 σ 本身不是 A 的定理．概括地说，接下去的讨论是这样的：如果 $A \vdash \sigma$，那么 σ 是假的，这与 A 是取值为真的句子集相矛盾．因此，σ 是取值为真的句子，但 $A \nvdash \sigma$．

为了构造 σ，我们首先考虑这样的一个三元关系 R：

$\langle a,b,c\rangle \in R$　iff　a 是某个公式 α 的哥德尔数，c 是从 $\alpha(\mathbf{S}^b\mathbf{0})$ 的 A 出发的某个推理 $\mathcal{G}$ 上的值

因为 $\{\sharp\alpha|\alpha \in A\}$ 在 $\mathfrak{N}$ 中可定义，因此 R 也是可定义的．(这里面的细节要等几节后才能证明．) 令 ρ 是 $\mathfrak{N}$ 中定义 R 的一个公式，q 为下面这个公式的哥德尔数

$$\forall v_3 \,\neg\, \rho(v_1, v_1, v_3).$$

(在此，用下列记号：$\varphi(t) = \varphi_t^{v_1}, \varphi(t_1,t_2) = (\varphi_{t_1}^{v_1})_{t_2}^{v_2}$，等等．) 令 σ 为

$$\forall v_3 \neg\, \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3).$$

则 σ 表示任何一个数都不是从 A 出发的某个推理 $\mathcal{G}$ 上的值，这个推理是将哥德尔数为 q 的公式中的变元 v_1 换成数字 q 后所得到的公式的推理；即任何一个数都不是 σ 的推论 $\mathcal{G}$ 的值．

假设和我们期望的结果相反，存在一个从 A 出发的 σ 的推论．我们设 k 是这个推论 $\mathcal{G}$ 的值，则 $\langle q,q,k\rangle \in R$ 并且

$$\models_{\mathfrak{N}} \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^k\mathbf{0}).$$

显然

$$\sigma \vdash \neg\, \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^k\mathbf{0})$$

并且这两个公式说明 σ 在 $\mathfrak{N}$ 中是假的．但 $A \vdash \sigma$ 和 A 中的句子在 $\mathfrak{N}$ 中都是真的，这就得到了矛盾．

因此，不存在从 A 出发 σ 的推论．这样对于每个 k，我们有 $\langle q,q,k\rangle \notin R$．即对于每个 k，

$$\models_{\mathfrak{N}} \neg\, \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^k\mathbf{0}),$$

由此我们有 (使用替换引理)

$$\models_{\mathfrak{N}} \forall v_3 \neg\, \rho(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3);$$

即，σ 在 $\mathfrak{N}$ 中是真的．■

后面我们将利用丘奇论题来证明，任意一个可判定的自然数集合在 $\mathfrak{N}$ 中都可定义．从
185
而得出结论，$\text{Th}\,\mathfrak{N}$ 是不可公理化的．

推论 30B 句子的哥德尔数集 $\{\#\tau | \models_{\mathfrak{N}} \tau\}$ 在 $\mathfrak{N}$ 中是不可以定义的，其中 τ 是在 $\mathfrak{N}$ 中真的句子.

证明 如果这个集合是可定义的，我们可以在上一个定理中取 $A = \text{Th}\,\mathfrak{N}$，这样就得到矛盾. ■

3.5 节中将采用自代入法，但与此处不同的是，句子 σ 要表达的是，“我是假的．”(这和著名的说谎者悖论有关！)

但如果你认为“自代入法”像是一个魔术把戏，那么，有另一个方法来说明这种情况：**对角线法**，它不使用明显的自代入.

我们从定义一个自然数上的二元关系 P 开始，

$$\langle a, b\rangle \in P \iff a \text{ 是公式 } \alpha(v_1)(\text{只有 } v_1 \text{ 是自由的}) \text{ 的哥德尔数，并且 } \models_{\mathfrak{N}} \alpha(\mathbf{S}^b\mathbf{0}).$$

(更通俗地说，$\langle a, b\rangle \in P \Leftrightarrow$“$a$ 说 b 是真的．”) 那么，任意一个在 $\mathfrak{N}$ 中可定义的自然数集合都等于，对某个 a，P 的一个“垂直剖面”：

$$P_a = \{b \mid \langle a, b\rangle \in P\}.$$

我们只要取 a 为定义该自然数集的公式的哥德尔数，然后再利用 $\models_{\mathfrak{N}} \alpha(\mathbf{S}^b\mathbf{0}) \Leftrightarrow \models_{\mathfrak{N}} \alpha(v_1)[\![b]\!]$ 即可.

所以任何一个 (在 $\mathfrak{N}$ 中) 可定义的自然数集合必然在序列 $P_1, P_2, \cdots$ 中．现在我们把这个序列“对角化”，定义集合：

$$H = \{b \mid \langle b, b\rangle \notin P\}.$$

(更通俗地说，$b \in H \Leftrightarrow$“b 说 b 不真．”) 那么 H 不在序列 $P_1, P_2, \cdots$ 中．(比如，$H \neq P_3$，因为 $3 \in H \Leftrightarrow 3 \notin P_3$，所以 3 属于且仅属于这两个集合中的一个．) 因此，H 在 $\mathfrak{N}$ 中不能定义.

为什么 H 是不可定义的呢？毕竟，我们在上面已经指出了

$$b \in H \iff \text{非 } [b \text{ 是公式 } \alpha(v_1)(\text{只有 } v_1 \text{ 是自由的}) \text{ 的哥德尔数，并且 } \models_{\mathfrak{N}} \alpha(\mathbf{S}^b\mathbf{0})].$$

是什么妨碍了将上面的定义翻译成代数语言？我们将证明公式的哥德尔数不是障碍，我们可以翻译它，v_1 是自由的以及将 $\mathbf{S}^b\mathbf{0}$ 代入公式也不是障碍．通过消去量词，我们将证明唯一可能的障碍就是：一个句子在 $\mathfrak{N}$ 中是真的.

定理 30C (a) 句子在 $\mathfrak{N}$ 中为真的哥德尔数的集合 $\{\sharp\tau | \models_{\mathfrak{N}} \tau\}$，在 $\mathfrak{N}$ 中是不可以定义的.

*(b) 理论 $\text{Th}\,\mathfrak{N}$ 是不可判定的.

*(c) 理论 $\text{Th}\,\mathfrak{N}$ 是不可公理化的. [186]

证明 (a) 和利用自代入法证明的推论 30B 是相同的，我们在上面已经概略地证明过了．即，如果我们假设它的反面成立，也就是 $\text{Th}\,\mathfrak{N}$ 在 $\mathfrak{N}$ 中可定义，那么上面的集合 H 也是可定义的．这是不可能的.

对于 (b)，我们只要证明每个可判定的自然数集一定在 $\mathfrak{N}$ 中可定义就可以了．如果 $\text{Th}\,\mathfrak{N}$ 可判定，那么对应的数集 $\{\sharp\tau | \models_{\mathfrak{N}} \tau\}$ 就是可判定的，进而在 $\mathfrak{N}$ 中可定义．这同样是不可能的.

由于 Th $\mathfrak{N}$ 是完全理论, (c) 可以从 (b) 和引理 26I 得到. ■

第三, *可计算性方法* 展示了"取值为真"和"可证明"两个概念之间的差别. 从 2.6 节我们知道, 当 A 是公理的可判定集 (甚至是能行可枚举集) 时, 我们可以为 Th $\mathfrak{N}$ 找到可证明句子的集合 Cn A, 它将是能行可枚举的.

与此形成对比的是, 可计算性方法和丘奇论题将证明所有取值为真的句子的集合 Th $\mathfrak{N}$ 不是能行可枚举的. 这个结果和定理 30C 有很密切的关系, 我们将在 3.6 节中运用对角线的方法来证明它.

***定理 30D** 对于任意一个公理的可判定集合 (甚至是能行可枚举集)A,

$$\text{Cn } A \neq \text{Th } \mathfrak{N}$$

因为等号左边的集合是能行可枚举的, 而右边的集合却不是.

定理 30D 提出了一个进退两难的问题: 要么从公理得到的推论是假的, 要么公理是不完全的, 从这个意义上说, 有取值为真的句子不能从那些公理推出.

可计算性方法在 3.5 节已经使用到, 但却在 3.6 节才正式提出, 在那与其他两种方法进行比较.

3.1 有后继数的自然数

我们从一种简单的情况开始考虑, 它虽然简单但能够解答我们提出的问题. 我们去掉参数集中的 $<, +, \cdot$ 和 $\mathbf{E}$, 只留下 $\forall, \mathbf{0}$ 和 $\mathbf{S}$, 那么, $\mathfrak{N}$ 相应的归约模型为

187

$$\mathfrak{N}_S = (\mathbb{N}; 0, S).$$

在这个归约语言中, 我们仍然能够描述自然数, 即 $\mathbb{N}$ 中的每一个元素. 但从算法的角度看, 在这个语言中我们能够表达的句子不够有趣.

对于 $\mathfrak{N}_S$, 与 $\mathfrak{N}$ 一样, 我们仍然要提出我们感兴趣的相同问题. 我们想知道理论集 Th $\mathfrak{N}_S$ 的复杂性; 想研究 $\mathfrak{N}_S$ 中的可定义性; 并且我们想综述 $\mathfrak{N}_S$ 的非标准模型.

为了研究带有后继函数的自然数理论 (Th $\mathfrak{N}_S$), 下面我们列出一些在 $\mathfrak{N}_S$ 中真的句子. (这些句子最终被证明是该理论的公理.)

S1. $\forall x \mathbf{S}x \neq \mathbf{0}$. 表示 0 不是任何数的后继.

S2. $\forall x \forall y(\mathbf{S}x = \mathbf{S}y \to x = y)$. 表示后继函数是一对一的.

S3. $\forall y(y \neq \mathbf{0} \to \exists x y = \mathbf{S}x)$. 表示任何一个非零数总有前驱.

S4.1 $\forall x \mathbf{S}x \neq x$.

S4.2 $\forall x \mathbf{SS}x \neq x$.

$\cdots$

S4.n $\forall x \mathbf{S}^n x \neq x$. 其中上标 n 表示函数 $\mathbf{S}$ 连续作用 n 次.

令 A_S 为包含 $\mathrm{S}_1, \mathrm{S}_2, \mathrm{S}_3, \mathrm{S}_{4.n} (n = 1, 2, \cdots)$ 的句子集. 显然, 这些句子在 $\mathfrak{N}_S$ 中是真的, 即 $\mathfrak{N}_S$ 是 A_S 的一个模型. 因此,

$$\text{Cn}A_S \subseteq \text{Th } \mathfrak{N}_S.$$

(即任意一个句子集，如果在 A_S 的每个模型中是真的，则在 $\mathfrak{N}_S$ 中也是真的．）然而，等号的成立却不是显然的，我们将在 A_S 的任意模型中证明这一点．

那么，公理 A_S 的任意模型 $\mathfrak{A}=(|\mathfrak{A}|;\mathbf{0}^{\mathfrak{A}},\mathbf{S}^{\mathfrak{A}})$ 是什么样的呢？根据 S_1,S_2 和 S_3 $\mathbf{S}^{\mathfrak{A}}$ 必须是从 $|\mathfrak{A}|$ 到 $|\mathfrak{A}|-\{\mathbf{0}^{\mathfrak{A}}\}$ 上的一对一映射．同时根据 S4.n，那么不存在不含长度为 n 的圈．因此，$|\mathfrak{A}|$ 必须包含“标准的”点：

$$\mathbf{0}^{\mathfrak{A}}\to\mathbf{S}^{\mathfrak{A}}(\mathbf{0}^{\mathfrak{A}})\to\mathbf{S}^{\mathfrak{A}}(\mathbf{S}^{\mathfrak{A}}(\mathbf{0}^{\mathfrak{A}}))\to\cdots,$$

这些都很明显，此处箭头表示 $\mathbf{S}^{\mathfrak{A}}$ 的作用．除了上述点外，模型中可能还包含其他的点．如果 $|\mathfrak{A}|$ 中包含另外一个点 a，那么 $|\mathfrak{A}|$ 还包含 a 的后继，后继的后继，等等．不仅如此，根据 S_3 我们知道，每一个非零元素都有前驱，并且这个前驱是唯一的 (根据 S_2)．因此，$|\mathfrak{A}|$ 中必须包含 a 的前驱，前驱的前驱，等等．这些元素都是互异的，否则就会出现有限的圈．因此，a 属于一个“Z 链”： 188

$$\cdots*\to*\to a\to\mathbf{S}^{\mathfrak{A}}(a)\to\mathbf{S}^{\mathfrak{A}}(\mathbf{S}^{\mathfrak{A}}(a))\to\cdots.$$

(我们称之为 Z 链是因为它们的分布就像整数集 $Z:\{\cdots,-1,0,1,2,\cdots\}$ 一样．)Z 链中的任何数都是存在的，但根据 S_2，任意两条 Z 链不能相交．类似地，任意一条 Z 链中都不能包含标准数．

我们可以用另外一种说法表达上面的观点：如果对 $|\mathfrak{A}|$ 中的某点 a 进行有限多次的 $S^{\mathfrak{A}}$ 作用后得到 b，我们称 a,b 是*等价的*．这个概念是一个等价关系．(自反性和对称性都是显然的；传递性可以从 $S^{\mathfrak{A}}$ 是一对一的这一事实得到．)$|\mathfrak{A}|$ 中的标准部分是包含 $\mathbf{0}^{\mathfrak{A}}$ 的等价类．对于 $|\mathfrak{A}|$ 中这个等价类以外的元素 a(如果有)，a 的等价类是 a 经由 $S^{\mathfrak{A}}$ 得到的所有元素以及所有能够由 $S^{\mathfrak{A}}$ 得到 a 的元素组成的集合．这个等价类就是上面所描述的一个 Z 链．

反之，如果 (该语言的) 任意一个结构 $\mathfrak{B}$ 中包含标准部分

$$\mathbf{0}^{\mathfrak{B}}\to\mathbf{S}^{\mathfrak{B}}(\mathbf{0}^{\mathfrak{B}})\to\mathbf{S}^{\mathfrak{B}}(\mathbf{S}^{\mathfrak{B}}(\mathbf{0}^{\mathfrak{B}}))\to\cdots$$

和 Z 链中的所有数组成的非标准部分，那么它就是 A_S 的模型．(我们可以逐条检验 A_S 中的公理，它们在 $\mathfrak{B}$ 中都是真的．) 这样，我们就对 A_S 的模型所具有的特点有了全面的了解．

如果 A_S 的模型 $\mathfrak{A}$ 有可数多条 Z 链，则 $|\mathfrak{A}|$ 是可数的．一般来说，若 Z 链集合的基数[1]是 λ，则 $|\mathfrak{A}|$ 中所有元素的个数为 $\aleph_0+\aleph_0\cdot\lambda$．按照基数运算的法则 (见 0 章)，这个数等于 $\aleph_0$ 和 λ 中的较大者．因此，

$$\operatorname{card}|\mathfrak{A}|=\begin{cases}\aleph_0 & \text{如果 }\mathfrak{A}\text{ 有可数多条 }Z\text{ 链}\\ \lambda & \text{如果 }\mathfrak{A}\text{ 有不可数多条 }Z\text{ 链}\end{cases}$$

引理 31A 若 $\mathfrak{A}$ 和 $\mathfrak{A}'$ 均为 A_S 的模型，且它们有相同数量的 Z 链，则它们同构．

证明 $\mathfrak{A}$ 和 $\mathfrak{A}'$ 的标准部分之间只存在唯一的同构．根据假设，$\mathfrak{A}$ 和 $\mathfrak{A}'$ 的 Z 链集合之间也有一个一一对应，即对于 $\mathfrak{A}$ 的每一条链都能找到 $\mathfrak{A}'$ 的一条链与之对应．显然任意两条 Z 链之间是同构的．综合上述各个独立的同构 (运用选择公理)，我们就得到了从 $\mathfrak{A}$ 到 $\mathfrak{A}'$ 上的同构． ■

1. 为了避免不可数基数的情况，见习题 3.

这样，在同构的意义下，A_S 的模型是由它的 Z 链的数目来决定的．对于 $\mathfrak{N}_S$ 来说，这个数目是零，但对于其他的模型，任何数目都是有可能的．

189

大家应该注意到，这个语言不能表达句子："不存在 Z 链．" 实际上，不存在句子集 Σ 使得，A_S 的模型 $\mathfrak{A}$ 满足 Σ 当且仅当 $\mathfrak{A}$ 不存在 Z 链．根据 LST 定理，存在不可数的结构 $\mathfrak{A}$ 使得 $\mathfrak{A} \equiv \mathfrak{N}_S$．但 $\mathfrak{A}$ 有不可数多条 Z 链，而 $\mathfrak{N}_S$ 没有．

定理 31B　设 $\mathfrak{A}$ 和 $\mathfrak{B}$ 均为 A_S 的不可数模型，且它们有相同的基数，则 $\mathfrak{A}$ 和 $\mathfrak{B}$ 同构．

证明　我们从上面的讨论中知道，$\mathfrak{A}$ 有 card $\mathfrak{A}$ 条 Z 链，$\mathfrak{B}$ 有 card $\mathfrak{B}$ 条 Z 链．由于 card $\mathfrak{A}$ = card $\mathfrak{B}$，所以它们有相同数目的 Z 链，因而是同构的．■

定理 31C　Cn A_S 是完全理论．

证明　使用 2.6 节中的洛斯 – 瓦特测试．上一个定理表明 Cn A_S 在不可数基数上是范畴的．进一步，由于 A_S 没有有限模型，运用洛斯 – 瓦特测试就可以得到该定理．■

推论 31D　Cn A_S = Th $\mathfrak{N}_S$．

证明　我们已知 Cn $A_S \subseteq$ Th $\mathfrak{N}_S$，而符号 $\subseteq$ 之前的理论是完全的，之后的理论是可满足的．■

***推论 31E**　Th $\mathfrak{N}_S$ 是可判定的．

证明　任意一个可公理化的完全理论是可判定的 (根据推论 25G)，而 A_S 恰好是这个理论可判定的公理集．■

量词消去

即使我们已经知道了一个理论是可判定的，但为其找到一个实际能行的判定过程仍然是非常吸引人的．我们将在 "量词消去" 的基础上，为 Th $\mathfrak{N}_S$ 找到这样一个过程．

定义　我们称理论 T 是实现量词消去的，当且仅当对于每一公式 φ，存在一个无量词公式 ψ，使得

$$T \vDash (\varphi \leftrightarrow \psi).$$

实际上，我们只需要考虑某些特殊形式的公式 ψ 的量词消去就足够了：

定理 31F　假设对于每个具有下列形式的公式 φ

$$\exists x(\alpha_0 \wedge \cdots \wedge \alpha_n),$$

其中每个 α_i 是原子公式或原子公式的否定，存在一个无量词公式 ψ 使得 $T \vDash (\varphi \leftrightarrow \psi)$，则 T 实现量词消去．

190

证明　首先我们断言，对于任意一个无量词公司 θ，我们可以找到与 $\exists x\theta$ 等价的无量词公式，可以找到与之等价的无量词公式．我们先把公式 θ 改写成析取范式 (根据推论 15C)：

$$\exists x[(\alpha_0 \wedge \cdots \wedge \alpha_m) \vee (\beta_0 \wedge \cdots \wedge \beta_n) \vee \cdots \vee (\xi_0 \wedge \cdots \wedge \xi_t)],$$

它逻辑等价于

$$\exists x(\alpha_0 \wedge \cdots \wedge \alpha_m) \vee \exists x(\beta_0 \wedge \cdots \wedge \beta_n) \vee \cdots \vee \exists x(\xi_0 \wedge \cdots \wedge \xi_t).$$

根据假设，上面这个公式中的每一析取项都可以用一个无量词公式替换.

对于任意的公式，利用这个结果不难得到与之等价的无量词公式. 我们把它留作习题 (见习题 2). ■

对于结构 $\mathfrak{A}$ 的理论 Th $\mathfrak{A}$ 来说，这个定义可以重述为：Th $\mathfrak{A}$ 实现量词消去，当且仅当对于每一个公式 φ，都存在一个无量词公式 ψ，使得 φ 和 ψ "在 $\mathfrak{A}$ 中等价"；即对于 $|\mathfrak{A}|$ 中任意变元的指派 s，都有

$$\models_{\mathfrak{A}} (\varphi \leftrightarrow \psi)[s].$$

定理 31G Th $\mathfrak{N}_S$ 实现量词消去.

证明 根据上面的定理，我们只需要考虑公式

$$\exists x(\alpha_0 \wedge \cdots \wedge \alpha_q),$$

的量词消去，其中 α_i 是原子公式或原子公式的否定. 我们将一步步找到与之等价的无量词公式. 实际上，与它等价的新公式是 A_S 的推论，见习题 3.

在 $\mathfrak{N}_S$ 的语言中，所有的项都有形式 $\mathbf{S}^k u$，其中 u 为 $\mathbf{0}$ 或变元. 原子公式都是等式. 我们假设变元 x 出现在每个 α_i 中. 因为如果 x 不在 α 中出现，则

$$\exists x(\alpha \wedge \beta) \models\!\!\!\dashv \alpha \wedge \exists x\beta.$$

这样，每个 α_i 都是下面的形式

$$\mathbf{S}^m x = \mathbf{S}^n u$$

或其否定，其中 u 为 $\mathbf{0}$ 或变元. 我们可以进一步假设 u 与 x 不相同，因为，当 $m = n$ 时，$\mathbf{S}^m x = \mathbf{S}^n x$ 可以用 $\mathbf{0} = \mathbf{0}$ 代替；当 $m \neq n$ 时，可用 $\mathbf{0} \neq \mathbf{0}$ 代替. 191

情形 1：每个 α_i 都是不等式. 则公式用 $\mathbf{0} = \mathbf{0}$ 代替. (为什么？)

情形 2：至少有一个 α_i 是等式，不妨设为 α_0：

$$\mathbf{S}^m x = t,$$

其中项 t 中不含 x. 由于 x 的值必须是非负的，因此我们用下面的公式代替 α_0：

$$t \neq \mathbf{0} \wedge \cdots \wedge t \neq \mathbf{S}^{m-1}\mathbf{0},$$

(如果 $m = 0$，则用 $\mathbf{0} = \mathbf{0}$ 替代). 对于其他的 α_j，我们先用

$$\mathbf{S}^{k+m} x = \mathbf{S}^m u,$$

代替

$$\mathbf{S}^k x = u$$

进而变成

$$\mathbf{S}^k t = \mathbf{S}^m u.$$

这样我们就得到了一个不含 x 的公式，因此量词被消去了. ■

在量词消去的过程中，有一些有趣的结果值得我们关注. 第一，我们得到了一种全新的方法来证明 Cn A_S 的完全性. 对于一个句子 σ，量词消去的过程给出了一个无量词句子 τ 使得 $A_S \vDash (\sigma \leftrightarrow \tau)$ (见习题 3). 我们可以断言：要么 $A_S \vDash \tau$，要么 $A_S \vDash \neg\tau$. 因为 τ 是由原子公式通过 $\neg$ 和 $\rightarrow$ 联结而成的. 而原子公式必然有形式 $\mathbf{S}^k\mathbf{0} = \mathbf{S}^l\mathbf{0}$，如果 $k = l$，则这个公式可由 A_S 推出；如果 $k \neq l$，则这个公式被 A_S 拒绝 (即不能从 A_S 推出). (实际上，证明这一点只需要 $\{\mathrm{S}_1, \mathrm{S}_2\}$ 就够了.) 由于每一个原子公式都能被推出或被拒绝，因此每个无量词句子也是如此，故我们的断言就是正确的. 因此，要么 $A_S \vDash \sigma$，要么 $A_S \vDash \neg\sigma$.

另一个结果是关于 $\mathfrak{N}_S$ 中的可定义性问题，具体见习题 4，5. 对于任意一个只含有 v_1 和 v_2 两个自由变元的公式 φ，我们可以找到一个无量词公式 ψ(含有相同的自由变元) 使得

$$\text{Th } \mathfrak{N}_S \vDash \forall v_1 \forall v_2 (\varphi \leftrightarrow \psi);$$

即

$$\vDash_{\mathfrak{N}_S} \forall v_1 \forall v_2 (\varphi \leftrightarrow \psi).$$

192 这样，由 φ 定义的关系也可以由无量词公式来定义.

习题

1. 设 A_S^* 是包含 $\mathrm{S}_1, \mathrm{S}_2$ 和所有具有以下形式的句子集合：

$$\varphi(\mathbf{0}) \rightarrow \forall v_1 (\varphi(v_1) \rightarrow \varphi(\mathbf{S}v_1)) \rightarrow \forall v_1 \varphi(v_1),$$

   其中 φ 是合式公式 (在 $\mathfrak{N}_S$ 的语言中)，φ 中只有 v_1 是自由变元. 证明 $A_S \subseteq \text{Cn } A_S^*$，进而 Cn $A_S^* =$ Th $\mathfrak{N}_S$. (此处 $\varphi(t)$ 即 $\varphi_t^{v_1}$. 上述式子被称作 φ 的*归纳公理*.)
2. 完成定理 31F 的证明. *提示*：用归纳法.
3. 给定一个公式 φ，对于 Th $\mathfrak{N}_S$ 量词消去的证明给出了如何找到与 φ 等价的无量词公式 ψ 的方法. 在不利用 Cn A_S 是完全的这个条件下，证明：$A_S \vDash (\varphi \leftrightarrow \psi)$. (这是证明 Cn$A_S$ 完全性的另一种方法，该方法不涉及 Z 链和洛斯 – 瓦特测试.)
4. 证明：$\mathbb{N}$ 的某个子集在 $\mathfrak{N}_S$ 中是可定义的，当且仅当它是有限的或者它在 $\mathbb{N}$ 中的补集是有限的.
5. 证明：序关系 $(\langle m, n\rangle | m < n, m, n \in \mathbb{N})$ 在 $\mathfrak{N}_S$ 中不可定义. *提示*：只需要证明不存在能够定义这个序关系的无量词公式. 如果关系 $R \subseteq \mathbb{N}\times\mathbb{N}$ 能被有限条直线覆盖，我们就称 R 是*线性的*. 如果 R 是线性关系的补集，我们称之为*余线性的*. 证明：任意在 $\mathfrak{N}_S$ 中可定义的关系或者是线性的或者是余线性的，并且序关系既不是线性的也不是余线性的.
6. 证明：Th $\mathfrak{N}_S$ 不能有限公理化. *提示*：只需要证明 A_S 的有限子集都不满足即可，然后运用 2.6 节中的结果.

3.2　数论的其他归约模型[1]

首先，我们在语言中加入符号 <，那么，现在所要讨论的结构是

193

$$\mathfrak{N}_L = (\mathbb{N}; 0, S, <).$$

和 Th $\mathfrak{N}_S$ 一样，我们要证明这个结构的理论是可判定的，也可以实现量词消去. 但与 Th $\mathfrak{N}_S$ 不同的是，它可以被有限公理化，并且在任意不可数基数上都不是范畴的.

1. 本节可以跳过.

我们将选取由以下 6 个句子组成的有限集合 A_L 作为 Th $\mathfrak{N}_L$ 的公理. $x \leqslant y$ 当然是 $(x < y \vee x = y)$ 的简写, $x \nleqslant y$ 是 $x = y$ 的否定.

$$\forall y \quad (y \neq \mathbf{0} \rightarrow \exists x y = \mathbf{S}x) \tag{S3}$$

$$\forall x \forall y \quad (x < \mathbf{S}y \leftrightarrow x \leqslant y) \tag{L1}$$

$$\forall x \quad x \nless \mathbf{0} \tag{L2}$$

$$\forall x \forall y \quad (x < y \vee x = y \vee y < x) \tag{L3}$$

$$\forall x \forall y \quad (x < y \rightarrow y \nless x) \tag{L4}$$

$$\forall x \forall y \forall z \quad (x < y \rightarrow y < z \rightarrow x < z) \tag{L5}$$

一方面, 我们很容易知道上述 6 条公理在 $\mathfrak{N}_L$ 中是真的, 这样, Cn $A_L \subseteq$ Th $\mathfrak{N}_L$. 另一方面, 反向的结论并不显然, 还需要证明. 我们先列出从上述公理得到的一些结论.

(1) $A_L \vdash \forall x x < \mathbf{S}x$.

证明 在 L1 中用 y 代替 x. ■

(2) $A_L \vdash \forall x x \nless x$.

证明 在 L4 中用 y 代替 x. ■

(3) $A_L \vdash \forall x \forall y (x \nless y \leftrightarrow y \leqslant x)$ (三分律).

证明 "→" 利用 L3. "←" 利用 L4 和 (2). ■

(4) $A_L \vdash \forall x \forall y (x < y \leftrightarrow \mathbf{S}x < \mathbf{S}y)$.

证明 从 A_L 我们可以得到以下的等价条件:

$$\begin{aligned} x < y &\leftrightarrow y \nleqslant x && \text{用 (3)} \\ &\leftrightarrow y \nless \mathbf{S}x && \text{用 L1} \\ &\leftrightarrow \mathbf{S}x \leqslant y && \text{用 (3)} \\ &\leftrightarrow \mathbf{S}x < \mathbf{S}y && \text{用 L1.} \end{aligned}$$

■

(5) $A_L \vdash$ S1 且 $A_L \vdash$ S2.

证明 S1 可以从 L2 和 (1) 得到. S2 可以利用 L3 和 (2) 从 (4) 中得到. ■

(6) $A_L \vdash$ S4.n , $n = 1, 2, \cdots$.

证明 利用 L5 从 (1), (2) 可以得到. ■ 194

这样, (当我们忽略 $<^{\mathfrak{A}}$ 时) 任意一个 A_L 的模型 $\mathfrak{A}$ 也是 A_S 的模型. 因此, 它一定包含标准部分以及若干个 Z 链 (Z 链的个数可以为零). 另外, 模型中有序关系 $<^{\mathfrak{A}}$.

定理 32A 理论 Cn A_L 实现量词消去.

证明 我们同样考虑公式

$$\exists x(\beta_0 \wedge \cdots \wedge \beta_p),$$

其中每个 β_i 是原子公式或原子公式的否定. 和 3.1 节一样, 项的形式只有 $\mathbf{S}^k u$, 其中 u 为 $\mathbf{0}$ 或变元. 原子公式有两种可能,

$$\mathbf{S}^k u = \mathbf{S}^l t \text{ 和 } \mathbf{S}^k u < \mathbf{S}^l t.$$

(1) 我们省略否定符号. 将 $t_1 \not< t_2$ 用 $t_2 < t_1 \vee t_1 = t_1$ 代替, $t_1 \neq t_2$ 用 $t_1 < t_2 \vee t_2 < t_1$ 来代替. (这由 L3 和 L4 保证.) 我们对原子公式进行重新分组, 并注意到

$$\exists x(\varphi \vee \psi) \models\!\!\dashv \exists x\varphi \vee \exists x\psi,$$

我们可以重新得到如下形式的公式

$$\exists x(\alpha_0 \wedge \cdots \wedge \alpha_q),$$

其中 α_i 是原子公式.

(2) 我们假设变元 x 在每个 α_i 中出现, 这是因为如果 x 不在 α 中出现, 则

$$\exists x(\alpha \wedge \beta) \models\!\!\dashv \alpha \wedge \exists x\beta.$$

进而, 我们可以假设 x 只出现在等式或不等式 α_i 的一边. 对于公式 $\mathbf{S}^k x = \mathbf{S}^l x$, 可以像 3.1 节中那样处理. 对于 $\mathbf{S}^k x < \mathbf{S}^l x$, 如果 $k < l$, 则该公式用 $\mathbf{0} = \mathbf{0}$ 来代替, 否则用 $\mathbf{0} \neq \mathbf{0}$ 来代替. (可由 L1 和 L4 得到.)

情形 1: 假设某个 α_i 为等式. 接下去的证明与定理 31G 中情形 2 的量词消去的证明过程相同.

情形 2: 每个 α_i 都是不等式. 则原公式可以写为

$$\exists x\Big(\bigwedge_i t_i < \mathbf{S}^{m_i} x \wedge \bigwedge_j \mathbf{S}^{n_j} x < u_j\Big).$$

(这里 $\bigwedge_i$ 表示以 i 为下标的公式的合取, 因此 $\gamma_0 \wedge \gamma_1 \wedge \cdots \wedge \gamma_k$ 可简写为 $\bigwedge_i \gamma_i$.) 在第一个合取式 $\bigwedge_i t_i < \mathbf{S}^{m_i} x$ 中, x 有下界; 在第二个合取式 $\bigwedge_j \mathbf{S}^{n_j} x < u_j$ 中, x 有上界. 如果第二个合取式不存在 (即 x 没有上界), 则原公式由 $\mathbf{0} = \mathbf{0}$ 来代替. (为什么?) 如果第一个合取式不存在 (即 x 没有下界), 则原公式由下式来代替

195

$$\bigwedge_j \mathbf{S}^{n_j} \mathbf{0} < u_j,$$

这说明 0 是满足上界的. 否则, 如果二者都存在, 我们将原公式逐步改写为

$$\exists x \bigwedge_{i,j} (t_i < \mathbf{S}^{m_i} x \wedge \mathbf{S}^{n_j} x < u_j) \tag{1}$$

$$\exists x \bigwedge_{i,j} (\mathbf{S}^{n_j} t_i < \mathbf{S}^{m_i + n_j} x < \mathbf{S}^{m_i} u_j) \tag{2}$$

$$\Big(\bigwedge_{i,j} \mathbf{S}^{n_j+1} t_i < \mathbf{S}^{m_i} u_j\Big) \wedge \bigwedge_j \mathbf{S}^{n_j} \mathbf{0} < u_j. \tag{3}$$

最后一个公式表示“任何一个下界加 1 后满足任意上界, 并且 0 满足任意上界. ”这表明上确界和下确界之间存在着 x 的解. 第二个公式保证了 x 的解是正的.

在每一种情形中, 我们都得到了所求公式的无量词等价形式. ■

推论 32B (a) Cn A_L 是完全的.

(b) Cn $A_L = \text{Th}\ \mathfrak{N}_L$.

*(c) Th $\mathfrak{N}_L$ 是可判定的.

证明 (a) 同样可以利用定理 31G 的证明中的的讨论. 因为 Cn $A_L \subseteq \text{Th}\ \mathfrak{N}_L$, 并且 Th $\mathfrak{N}_L$ 是可满足的, 因此 (b) 可以从 (a) 得到. 对于 (c), 我们可以利用已有的结果: 任意可公理化的完全理论是可判定的. 但量词消去的证明给出了更可行的判定过程. ■

推论 32C ℕ 的子集在 $\mathfrak{N}_L$ 中是可定义的, 当且仅当它是有限的或者补集是有限的.

证明 参照上一节的习题 4. ■

另一方面, $\mathfrak{N}_L$ 比 $\mathfrak{N}_S$ 有更多可定义的二元关系. 比如在上节的习题 5 中, 我们证明了序关系 $\{\langle m,n\rangle | m < n\}$ 在 $\mathfrak{N}_S$ 中不能定义.

推论 32D 加法关系, $\{\langle m,n,p\rangle | m+n=p\}$, 在 $\mathfrak{N}_L$ 中不能定义. 196

证明 如果我们能够定义加法, 就能定义偶数集合. 而偶数集合既不是有限的, 也不是补集有限的. ■

现在, 我们在语言中再加入加法符号 +, 则考虑的结构为

$$\mathfrak{N}_A = (\mathbb{N}; 0, S, <, +).$$

我们会简要地证明, 这个结构的理论也是可判定的. 但为了避免繁琐, 我们不再列出这个理论的公理集了.

Th $\mathfrak{N}_A$ 的非标准模型必然也是 Th $\mathfrak{N}_L$ 的模型. 因此, 它们也有标准部分, 以及一些 Z 链, 但 Z 链间的大小顺序不再是任意的了. 设 $\mathfrak{A}$ 是 Th $\mathfrak{N}_A$ 的非标准模型, 序关系 $<^{\mathfrak{A}}$ 诱导出 Z 链集合中良定义的序关系 (见习题 3). 我们断言: 不存在最大的 Z 链, 也不存在最小的 Z 链, 并且在任意两个 Z 链之间一定存在着另外一个 Z 链. 概括地说, 其原因主要是: 如果 a 属于某个 Z 链 (即 a 是 $\mathfrak{A}$ 的一个无限元), 则 $a +^{\mathfrak{A}} a$ 必然位于一个更大的 Z 链中. 而且一定存在一个 b 使得 $b +^{\mathfrak{A}} b$ 是 a 或 a 的后继; b 一定位于一个更小的 Z 链中. 若 a_1, a_2 属于不同的 Z 链, 则一定存在一个 b 使得 $b +^{\mathfrak{A}} b$ 是 $a_1 +^{\mathfrak{A}} a_2$ 或它的后继. 这样, b 即位于 a_1, a_2 的 Z 链之间的 Z 链中. (这段理由似乎很难理解. 有兴趣研究无穷大数的读者可以试着补充其中的细节.)

***定理 32E(Presburger, 1929)** 结构 $\mathfrak{N}_A = (\mathbb{N}; 0, S, <, +)$ 的理论是可判定的.

这个定理的证明也是建立在量词消去的基础上. $\mathfrak{N}_A$ 理论本身不能实现量词消去, 比如, 定义偶数集的公式

$$\exists y\, v_1 = y + y$$

不等价于任何无量词公式. 为了克服这个困难, 我们增加一个新的符号 $\equiv_2$, 表示模 2 的同余关系. 类似地, 我们增加 $\equiv_3$, $\equiv_4$, $\cdots$. 那么, 这个扩展语言的结构为

$$\mathfrak{N}^{\equiv} = (\mathbb{N}; 0, S, <, +, \equiv_2, \equiv_3, \cdots),$$

其中 $\equiv_k$ 表示模 k 的二元同余关系. 事实表明, 该结构的理论的确满足量词消去.

197 虽然这并不意味着任意结构的理论都是可判定的，但毕竟我们可以从 任意 一个结构出发，一步一步添加新的关系，直到添加后的结构实现量词消去为止. 为了证明可判定性，对于给定的句子 σ，我们必须 (1) 能行地找到一个无量词公式 σ', (2) 判定 σ' 是否是真的.

现在，我们要给出 Th $\mathfrak{N}^{\equiv}$ 的量词消去过程. 对于一个项 t 和自然数 n, nt 表示 n 个 t 相加，即项 $t+t\cdots+t$, $0t$ 为 $\mathbf{0}$. 那么，任意一个项可以写成以下形式

$$\mathbf{S}^{n_0}\mathbf{0}+n_1x_1+\cdots+n_kx_k,$$

其中 $k\geqslant 0$, $n_i\geqslant 0$(x_i 是变元). 例如，

$$\mathbf{S}(x+\mathbf{S0})+\mathbf{S}y$$

可以写成

$$\mathbf{S}^3\mathbf{0}+x+y.$$

按照惯例，我们仍从公式 $\exists y(\beta_1\wedge\cdots\wedge\beta_n)$ 开始，其中 β_i 是原子公式或原子公式的否定.

(1) 消去否定号. 我们用 $(t_1<t_2\vee t_2<t_1)$ 代替 $\neg(t_1=t_2)$，用 $(t_1=t_2\vee t_2<t_1)$ 代替 $\neg(t_1<t_2)$. 同时用

$$t_1\equiv_m t_2+\mathbf{S}^1\mathbf{0}\vee\cdots\vee t_1\equiv_m t_2+\mathbf{S}^{m-1}\mathbf{0}.$$

代替 $\neg(t_1\equiv_m t_2)$. 则原公式可以重组为以下形式

$$\exists y(\alpha_1\wedge\cdots\wedge\alpha_m),$$

其中 α_i 为原子公式. 和前面一样，我们可以进一步假设，y 在每一个 α_i 中出现，且实际上 α_i 必为下列 4 种形式之一：

$$\begin{aligned} ny+t &= u,\\ ny+t &\equiv_m u,\\ ny+t &< u,\\ u &< ny+t, \end{aligned}$$

其中 u 和 t 为不含 y 的项. 在下面的叙述中，我们将用减号改写上面的公式：

$$\begin{aligned} ny &= u-t,\\ ny &\equiv_m u-t,\\ ny &< u-t,\\ u-t &< ny. \end{aligned}$$

198 这仅仅是利用减号移动了某些项的位置而得到的.

例如，我们有下面的公式

$$\exists y(w<4y\wedge 2y<u\wedge 3y<v\wedge y\equiv_3 t),$$

其中 t,u,v 和 w 是不含 y 的项.

(2) 统一 y 的系数. 设 p 是 y 的系数的最小公倍数. 无论是等式还是不等式, 通过乘以适当的因子, 每个原子公式中 y 的系数都可变为 p. 对于同余关系, 我们必须记住:

$$a \equiv_m b \quad \text{iff} \quad ka \equiv_{km} kb.$$

在上面的例子中, p 为 12, 我们有

$$\exists y(3w < 12y \wedge 12y < 6u \wedge 12y < 4v \wedge 12y \equiv_{36} 12t).$$

(3) 消去 y 的系数. 用 x 代替 py, 同时增加一个新的合取项 $x \equiv_p \mathbf{0}$. (不用 $\exists y \cdots 12y \cdots$ 而等价有: 存在 12 的倍数 x 使得 $\cdots x \cdots$. ") 这样, 我们的例子进一步转化为

$$\exists x(3w < x \wedge x < 6u \wedge x < 4v \wedge x \equiv_{36} 12t \wedge x \equiv_{12} \mathbf{0}).$$

(4) 特殊情况. 如果有一个原子公式是等式 $x + t = u$, 那么我们就用

$$\theta^x_{u-t} \wedge t \leqslant u.$$

代替 $\exists x\theta$. 这里用 "$u - t$" 来代替 x 是很自然的, 我们通过移项保证减号出现. 例如,

$$(x \equiv_m v)^x_{u-t} \text{ 即 } u \equiv_m v + t.$$

(5) 从现在起, 我们可以保证 $=$ 不出现. 因此, 我们有下面形式的公式

$$\begin{aligned}\exists x\,[&r_0 - s_0 < x \wedge \cdots \wedge r_{l-1} - s_{l-1} < x\\ &\wedge x < t_0 - u_0 \wedge \cdots \wedge x < t_{k-1} - u_{k-1}\\ &\wedge x \equiv_{m_0} v_0 - w_0 \wedge \cdots \wedge x \equiv_{m_{n-1}} v_{n-1} - w_{n-1}],\end{aligned}$$

其中 r_i, s_i, t_i, u_i, v_i 和 w_i 为不含 x 的项. 这个公式可以简写为

$$\exists x\Big[\bigwedge_{j<l} r_j - s_j < x \wedge \bigwedge_{i<k} x < t_i - u_i \wedge \bigwedge_{i<n} x \equiv_{m_i} v_i - w_i\Big].$$

如果没有同余等价类 (即 $n = 0$), 则上式表明在上界和下界之间存在一个非负的元素. 我们用下面的无量词公式来代替上式: 199

$$\bigwedge_{i<k}\bigwedge_{j<l}(r_j - s_j) + \mathbf{S0} < t_i - u_i \wedge \bigwedge_{i<k} \mathbf{0} < t_i - u_i.$$

设 M 是模 $m_0, \cdots, m_{n-1}$ 的最小公倍数, 则 $a + M \equiv_{m_i} a$. 因此, 随着 a 的增大, a 模 $m_0, \cdots, m_{n-1}$ 的余数形式有周期 M. 这样, 为了找到同余等价类的解, 我们只需要在 M 个连续的整数中寻找就够了.

现在就有一个公式表达, 存在一个自然数, 它不小于某些下界 $L_1, \cdots, L_l$, 且满足某些上界和某些同余等价类. 如果这个数存在, 则必是下列数之一:

$$\begin{aligned}&L_1, L_1 + 1, \cdots, L_1 + M - 1,\\ &L_2, L_2 + 1, \cdots, L_2 + M - 1,\\ &\cdots\\ &L_l, L_l + 1, \cdots, L_l + M - 1,\\ &0, 1, \cdots, M - 1.\end{aligned}$$

(考虑到每个 L_j 是负数的情况，我们添加了最后一行．为了避免把这一行作为特例，我们增加了一个新的下界 0．即令 $r_l = \mathbf{0}, s_l = \mathbf{S0}$ 使得

$$r_l - s_l < x$$

是公式 $\mathbf{0} < x + \mathbf{S0}$，这保证 x 是非负的．现在我们有了 $l+1$ 个下界．)

我们的公式 (保证 x 的解的存在性) 现在可以改写成一个无量词的析取式，这个式子说明上面矩阵中的一个数是非负解．

$$\begin{aligned}\bigvee_{j\leqslant l}\ \bigvee_{1\leqslant q\leqslant M}\ \Big[&\bigwedge_{i\leqslant l} r_i - s_i < (r_j - s_j) + \mathbf{S}^q\mathbf{0}\\ &\wedge \bigwedge_{i<k}(r_j - s_j) + \mathbf{S}^q\mathbf{0} < t_i - u_i\\ &\wedge \bigwedge_{i<n}(r_j - s_j) + \mathbf{S}^q\mathbf{0} \equiv_{m_i} v_i - w_i\Big].\end{aligned}$$

在给 x 增加了新的下界之后，上面的例子可以进一步写成：

$$\exists x(3w < x \wedge \mathbf{0} < x + \mathbf{S0} \wedge x < 6u \wedge x < 4v \wedge x \equiv_{36} 12t \wedge x \equiv_{12} \mathbf{0}).$$

200　无量词等价式是 72 个合取式的析取，每个合取式有 6 个组成部分．

到此，我们证明了定理的一半．上述过程告诉我们，如果给定句子 σ，怎样能行地找到一个无量词公式 τ (在语言 $\mathfrak{N}^{\equiv}$ 中)，使得 τ 是真的 (在考虑的结构中) 当且仅当 σ 是真的．下面我们要判定 τ 是否是真的．

这实际上很简单，我们只需要考虑原子句．任何一个无变元的项可以用 $\mathbf{S}^n\mathbf{0}$ 来表示．例如，

$$\mathbf{S}^n\mathbf{0} \equiv_m \mathbf{S}^p\mathbf{0}$$

是真的当且仅当 $n \equiv_m p$．■

这样，我们给出了 Th $\mathfrak{N}_A$ 的判定过程．然而，1974 年 Michael Fischer 和 Michael Rabin 证明了：对于太长的公式，足够快并且能行的判定过程是不存在的．

对于某个自然数集合 D，如果对于某个正数 p，使得对于任意的数 n，n 属于 D 当且仅当 $n+p$ 属于 D，则我们称 D 是周期的．称 D 是终周期的当且仅当如果存在一个正数 M 和 p 使得，对于所有比 M 大的数 n，$n \in D$ 当且仅当 $n+p \in D$．

定理 32F　一个自然数集合在 $(\mathbb{N}; 0, S, <, +)$ 中是可定义的当且仅当它是终周期的．

证明　习题 1 证明了每个终周期的集合都是可定义的．另一方面，我们假设 D 是可定义的，则 D 在 $\mathfrak{N}^{\equiv}$ 中可以由一个无量词公式定义 (公式中只有唯一一个变元 v_1)．由于所有终周期的集合在并、交和补运算下是封闭的，因此，我们只需要证明：在只含有变元 v_1 的 $\mathfrak{N}^{\equiv}$ 的语言中，每个原子公式都定义了一个终周期集合．我们要考虑下列 4 种可能：

$$\begin{aligned} nv_1 + t &= u,\\ nv_1 + t &< u,\\ u &< nv_1 + t,\\ nv_1 + t &\equiv_m u,\end{aligned}$$

其中 u 和 t 是数字. 前两个公式定义了有限集合 (其终周期为 1), 第 3 个公式定义一个有有限补集的集合, 最后一个公式则定义了一个周期为 m 的周期集合. ■

推论 32G 乘法关系 $\{\langle m,n,p\rangle | p = m\cdot n, p,m,n\in\mathbb{N}\}$ 在 $(\mathbb{N};0,S,<,+)$ 中不是可定义的.

证明 如果我们能够定义乘法, 那么就能利用它来定义平方数的集合. 但平方数的集合不是终周期的. ■ 201

习题

1. 证明: 任意自然数的终周期集合在结构 $\mathfrak{N}_A$ 中是可定义的.
2. 证明: 在结构 $(\mathbb{N};+)$ 中, 下列关系是可定义的.
   (a) 序关系, $\{\langle m,n\rangle | m<n\}$.
   (b) 零, $\{0\}$.
   (c) 后继, $\{\langle m,n\rangle | n=S(m)\}$.
3. 设 $\mathfrak{A}$ 是 $\mathrm{Th}\,\mathfrak{N}_L$ 的模型 (或 A_L 的模型). 对于 $|\mathfrak{A}|$ 中的元素 a 和 b, 定义等价关系:

   $$a\sim b \Leftrightarrow a,b \text{ 中的一个通过 } \mathbf{S}^{\mathfrak{A}} \text{ 的有限次作用之后能成为另一个}.$$

   令 $[a]$ 表示 a 所在的等价类, 则等价类间的序关系如下定义:

   $$[a]\prec[b] \quad \text{iff} \quad a<^{\mathfrak{A}} b \text{ 和 } a\nsim b.$$

   证明: 这个等价类集合上的序关系是良定义的.
4. 证明: 带有通常序关系的实数理论 $\mathrm{Th}(\mathbb{R};<)$ 实现量词消去. (假设语言包含等号.)

3.3 数论的子理论

我们现在回到 3.0 节中所述的数论完全语言, 语言中的参数有 $\forall$, $\mathbf{0}$, $\mathbf{S}$, $<$, $+$, $\cdot$ 和 $\mathbf{E}$. 这个语言的结构为

$$\mathfrak{N}=(\mathbb{N};0,S,<,+,\cdot,E).$$

实际上, 在 $(\mathbb{N};\cdot,E)$ 中我们可以定义 $\{0\},S,<$ 和 $+$. (见习题 1.) 在 3.8 节中我们将证明, 在 $(\mathbb{N};+,\cdot)$ 中, 除了可以定义 $0,S$ 和 $<$ 之外, 我们同样可以定义 E. 因此, 我们可以省略其中的一些参数. 但我们保留所有的参数 (特别是 $\mathbf{E}$) 是为了简化后面的一些证明.

我们将要看到, $\mathrm{Th}\,\mathfrak{N}$ 是一个非常强的理论, 它既不可判定也不可公理化. 为了证明这一事实 (及一些相关结果), 我们首先有针对性地选取了 $\mathrm{Th}\,\mathfrak{N}$ 的一个子理论, 它是可以有限公理化的. 正如 3.0 节中所示, 这个子理论必须能够给出可判定集. 我们选取的这个子理论是 $\mathrm{Cn}\,A_E$, 这里 A_E 包含下列 11 个句子的集合. (同前一节一样, $x<y\vee x=y$ 简写为 $x\leqslant y$.) 202

3.3.1 公理集 A_E

$$\forall x \quad \mathbf{S}x\neq\mathbf{0} \tag{S1}$$

$$\forall x\forall y \quad (\mathbf{S}x=\mathbf{S}y\rightarrow x=y) \tag{S2}$$

$$\forall x\forall y \quad (x<\mathbf{S}y\leftrightarrow x\leqslant y) \tag{L1}$$

$$\forall x \quad (x\nless\mathbf{0}) \tag{L2}$$

$$\forall x \forall y \quad (x < y \vee x = y \vee y < x) \tag{L3}$$

$$\forall x \quad x + \mathbf{0} = x \tag{A1}$$

$$\forall x \forall y \quad x + \mathbf{S}y = \mathbf{S}(x + y) \tag{A2}$$

$$\forall x \quad x \cdot \mathbf{0} = \mathbf{0} \tag{M1}$$

$$\forall x \forall y \quad x \cdot \mathbf{S}y = x \cdot y + x \tag{M2}$$

$$\forall x \quad x\mathbf{E0} = \mathbf{S0} \tag{E1}$$

$$\forall x \forall y \quad x\mathbf{ES}y = x\mathbf{E}y \cdot x \tag{E2}$$

由于 $\mathfrak{N}$ 是 A_E 的模型，因此 $\mathrm{Cn}\, A_E \subseteq \mathrm{Th}\, \mathfrak{N}$. 但此处的等号不成立 (我们将在 3.5 节中证明). 实际上，不成立的原因是 $A_E \not\vdash$S3，其中 S3 是句子 $\forall y(y \neq \mathbf{0} \to \exists x y = \mathbf{S}x)$.

前 5 个句子给出了一些关于 $\mathbf{S}$ 和 $<$ 的公理，它们在上一节中是很有用的．但这些并不是全部的公理．剩余的 6 条公理是描述加法，乘法和幂乘的 "递归" 公式．

我们首先证明 $\mathrm{Th}\, \mathfrak{N}$ 中的某些简单句子可以从 A_E 中推出．

引理 33A (a) $A_E \vdash \forall x x \not< \mathbf{0}$.

(b) 对于任意自然数 k,

$$A_E \vdash \forall x(x < \mathbf{S}^{k+1}\mathbf{0} \leftrightarrow x = \mathbf{S}^0\mathbf{0} \vee \cdots \vee x = \mathbf{S}^k\mathbf{0}).$$

我们注意到，(a) 可以看作情形 (b) 中当 $k = -1$ 时的特例，若其中没有析取项则取为 $\perp$. 这个引理告诉我们 A_E "知道" 小于 k 的数有多少个，比如，它知道小于 7 的数是 0,1,2,3,4,5,6. 因此，在 A_E 的任意模型中，标准点——能够用 $S^k\mathbf{0}$ 表示的数——以自然的方式排列．如果存在无穷大点，则无穷大点 (L3) 大于任何一个标准点．

证明 (a) 就是 L2. 为了证明 (b)，我们对 k 进行归纳．我们有下面的式子作为 L1 的结果：

[203]

$$x < \mathbf{S0} \leftrightarrow x < \mathbf{0} \vee x = \mathbf{0},$$

再加上 L2，我们有

$$x < \mathbf{S0} \leftrightarrow x = \mathbf{0},$$

这是情形 (b) 中当 $k = 0$ 的情况．在接下去的归纳中，我们仍然利用 L1:

$$x < \mathbf{S}^{k+1}\mathbf{0} \leftrightarrow x < \mathbf{S}^k\mathbf{0} \vee x = \mathbf{S}^k\mathbf{0}.$$

根据归纳假设，$x < \mathbf{S}^k\mathbf{0}$ 可以用下面的公式代替：

$$x = \mathbf{S}^0\mathbf{0} \vee \cdots \vee x = \mathbf{S}^{k-1}\mathbf{0},$$

这样就能得到 (b). ■

引理 33B 对于任意无变元项 t，存在唯一的自然数 n，使得

$$A_E \vdash t = \mathbf{S}^n\mathbf{0}.$$

证明 唯一性是显然的．(为什么？因为只能存在一个 n，使得 $t = \mathbf{S}^n\mathbf{0}$ 在 $\mathfrak{N}$ 中为真．) 为了证明存在性，我们对 t 进行归纳．如果 t 为 $\mathbf{0}$，我们取 $n = 0$. 如果 t 为 $\mathbf{S}u$，那么根据归纳假设，对于某个 m，有 $A_E \vdash u = \mathbf{S}^m\mathbf{0}$，进而，$A_E \vdash t = \mathbf{S}^{m+1}\mathbf{0}$.

现在假设 t 是 u_1+u_2. 根据归纳假设, 对于 m 和 n, 有 $A_E \vdash t = \mathbf{S}^m\mathbf{0}+\mathbf{S}^n\mathbf{0}$. 我们现在反复利用 A2 n 次, 最后利用 1 次 A1, 就得到 $A_E \vdash t = \mathbf{S}^{m+n}\mathbf{0}$. 乘法和幂乘的证明类似. ■

作为这个引理的特例,"2+2=4" (即 $\mathbf{S}^2\mathbf{0}+\mathbf{S}^2\mathbf{0}=\mathbf{S}^4\mathbf{0}$) 就是 A_E 的结果. A_E 至少能够计算出无变元项的值, 当然, 上面的证明说明的问题不止这些. 对于无变元项 t, 它还给出了精确的方法去能行地寻找唯一的 n 使得 $A_E \vdash t = \mathbf{S}^n\mathbf{0}$.

定理 33C 对于任意在 $\mathfrak{N}$ 中为真的无量词句子 τ, $A_E \vdash \tau$.

证明 见习题 2. 从原子公式开始考虑, 对于无变元项 t_1 和 t_2, 原子公式具有形式 $t_1 = t_2$ 或 $t_1 < t_2$. 证明如果 τ 在 $\mathfrak{N}$ 中为真, 则 A_E 能推出 τ; 如果 τ 在 $\mathfrak{N}$ 中为假, 则 A_E 拒绝 τ(即推出 $\neg\tau$). ■

接下来, 通过允许 τ 含有"有界量词", 我们进一步推广了定理 33C; 见定理 33I.

我们将采用一种简单的记法来表示替换 (在 2.7 节中已经用过), 这在接下来的叙述中是非常有帮助的:

$$\varphi(t) = \varphi_t^{v_1},$$
$$\varphi(t_1,t_2) = (\varphi_{t_1}^{v_1})_{t_2}^{v_2},$$

等等. 这里, $\varphi = \varphi(v_1) = \varphi(v_1,v_2)$. 通常替换项为数字, 例如　[204]

$$\varphi(\mathbf{S}^a\mathbf{0},\mathbf{S}^b\mathbf{0}) = (\varphi_{\mathbf{S}^a\mathbf{0}}^{v_1})_{\mathbf{S}^b\mathbf{0}}^{v_2}.$$

但有时我们也用其他的项代入, 如, $\varphi(x) = \varphi_x^{v_1}$, 其中 x 是变元. 然而, 如果 φ 中的 v_1 不能用 x 替换, 我们就必须取 $\varphi(x) = \psi_x^{v_1}$, 其中 ψ 是 φ 合适的字母变换式.

在下面的证明中 (本章的另一处) 我们将用到 2.5 节中替换引理的结果: 设公式 φ 至多有 $v_1,\cdots,v_n$ 是自由变元, $a_1,\cdots,a_n$ 是自然数, 则

$$\models_{\mathfrak{N}} \varphi[\![a_1,\cdots,a_n]\!] \Leftrightarrow \models_{\mathfrak{N}} \varphi(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_n}\mathbf{0}).$$

一个存在 ($\exists_1$) 公式是指具有形式如 $\exists x_1\cdots\exists x_k\theta$ 的公式, 其中 θ 为无量词公式. 下面的结果是定理 33C 的推广:

推论 33D 若 τ 是 $\mathfrak{N}$ 中为真的存在句, 则 $A_E \vdash \tau$.

证明 如果 $\exists v_1\exists v_2\theta$ 在 $\mathfrak{N}$ 中为真, 那么存在自然数 m,n, 使得 $\theta(\mathbf{S}^m\mathbf{0},\mathbf{S}^n\mathbf{0})$ 在 $\mathfrak{N}$ 中为真. 由于这是个无量词的真句子, 因此, 它可从 A_E 中推出. 但 A_E 在逻辑上可以一步步推出 $\exists v_1\exists v_2\theta$. ■

另一方面, 我们知道存在为真的全称 ($\forall_1$) 公式 (即形式 $\forall x_1,\cdots,\forall x_k\theta$, θ 为无量词), 它们不在 Cn A_E 中.

3.3.2 可表示关系

设 R 是 $\mathbb{N}$ 上的 m 元关系, 即 $R\subseteq\mathbb{N}^m$. 我们已经知道, 公式 ρ(其中只有 $v_1,\cdots,v_m$ 是自由变元) 在 $\mathfrak{N}$ 中定义 R 当且仅当对于 $\mathbb{N}$ 中任意的 $a_1,\cdots,a_m$,

$$\langle a_1,\cdots,a_m\rangle\in R\Leftrightarrow\models_{\mathfrak{N}}\rho[\![a_1,\cdots,a_m]\!]$$
$$\Leftrightarrow\models_{\mathfrak{N}}\rho(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0}).$$

(后两个条件等价是根据替换引理.) 我们可以把上述结果写成两个蕴涵式:

$$\langle a_1,\cdots,a_m\rangle\in R\Rightarrow\models_{\mathfrak{N}}\rho(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0}).$$
$$\langle a_1,\cdots,a_m\rangle\notin R\Rightarrow\models_{\mathfrak{N}}\neg\rho(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0}).$$

如果在这两个蕴涵式中,"在 $\mathfrak{N}$ 中为真"的概念可以被更强的概念"从 A_E 中推出"取代,我们称 ρ 在理论 Cn A_E 中可以表示 R.

一般地说,设 T 是含有 $\mathbf{0}$ 和 $\mathbf{S}$ 语言中的任意一个理论,则 ρ 在 T 中表示 R 当且仅当对于 $\mathbb{N}$ 中每一个 $a_1,\cdots,a_m$:

$$\langle a_1,\cdots,a_m\rangle\in R\Rightarrow\rho(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0})\in T,$$
$$\langle a_1,\cdots,a_m\rangle\notin R\Rightarrow(\neg\rho(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0}))\in T.$$

205

例如,ρ 在理论 Th $\mathfrak{N}$ 中表示 R 当且仅当 ρ 在 $\mathfrak{N}$ 中定义 R. 但 ρ 在 Cn A_E 中表示 R 当且仅当对于所有的 $a_1,\cdots,a_m$:

$$\langle a_1,\cdots,a_m\rangle\in R\Rightarrow A_E\vdash\rho(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0}),$$
$$\langle a_1,\cdots,a_m\rangle\notin R\Rightarrow A_E\vdash\neg\rho(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0}).$$

比如,$\mathbb{N}$ 上的等于关系在 Cn A_E 中可以用公式 $v_1=v_2$ 表示. 对于

$$m=n\Rightarrow\vdash\mathbf{S}^m\mathbf{0}=\mathbf{S}^n\mathbf{0},$$
$$m\neq n\Rightarrow\{\text{S1},\text{S2}\}\vdash\neg\,\mathbf{S}^m\mathbf{0}=\mathbf{S}^n\mathbf{0}.$$

一个关系在 T 中是*可表示的*,当且仅当存在一个公式,它在 T 中表示该关系.

我们有必要比较一下可表示和可定义这两个概念. 在这两个概念中,我们都是用公式来描述自然数之间的关系. 对于可定义性,我们关心的是句子的解释是否是真的;而对于在 Cn A_E 中可表示,我们关心的是句子是否能从公理*推出*.

设公式 φ 中只有 $v_1,\cdots,v_m$ 是自由变元,我们称 φ 由 A_E*数字确定*,当且仅当对于任意的自然数 $a_1,\cdots,a_m$,或者

$$A_E\vdash\varphi(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0})$$

或者

$$A_E\vdash\neg\varphi(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0})$$

定理 33E 公式 ρ 在 Cn A_E 中表示关系 R 当且仅当

(1) ρ 由 A_E 数字确定,且

(2) ρ 在 $\mathfrak{N}$ 中定义 R.

证明 首先我们要明确的事实是,$\mathfrak{N}$ 是 A_E 的模型. 如果 ρ 在 Cn A_E 中表示 R,则 (1) 显然成立;由于"$A_E\vdash$"蕴含"$\models_{\mathfrak{N}}$",因此,(2) 成立. 另一方面,如果 (1) 和 (2) 成立,则有

$\langle a_1,\cdots,a_m\rangle\in R\Rightarrow\models_{\mathfrak{N}}\rho(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0})$	根据 (2)
$\Rightarrow A_E\nvdash\neg\rho(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0})$	因为 $\mathfrak{N}$ 是 A_E 的模型
$\Rightarrow A_E\vdash\rho(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0})$	根据 (1)

对于 R 和 $\neg\rho$ 的补，情况类似. ■

3.3.3 丘奇论题

我们现在关注可表示性和可判定性之间的关系.

***定理 33F** 设 T 是一个可公理化的和谐理论，R 在 T 中可表示，则 R 是可判定的. 206

证明 设 ρ 在可公理化的和谐理论 T 中表示 R，我们已经知道 T 是能行可枚举的 (推论 25F). 则 R 的判定过程如下：

设 $a_1, \cdots, a_m$ 是 T 中元素的枚举. 如果 $\rho(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_m}\mathbf{0})$ 在这一枚举中，则 $\langle a_1, \cdots, a_m\rangle \in R$，判定过程结束；若 $\neg\rho(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_m}\mathbf{0})$ 在这一枚举中，则 $\langle a_1, \cdots, a_m\rangle \notin R$，判定过程结束.

根据可表示性，一个句子或其否定总会出现，因此，判定过程会结束. 因为 T 是和谐的，所以由判定过程给出的答案是正确的. ■

***推论 33G** 设 T 是一个可有限公理化的和谐理论，则任意在该理论中可表示的关系都是可判定的.

上述推论的逆命题是什么？我们不能证明该推论的逆命题，因为可判定性的概念是不正规的. 我们所掌握的不正规方法只能够给出可判定关系类的下界（即证明某些关系是可判定的)，却不能给出上界 (即证明不可判定性).

不过，我们还是有必要做一些有用的讨论来支持这个逆命题的正确性. 这将在学习完 3.4 节之后进行，那时讨论要比现在容易些. 概括地说，其中的主要思想是，我们能从有限的公理中找出判定过程所要的 (有限长) 指令.

上面的推论及推论的逆命题被称为*丘奇论题*. 这一论题与其说是真正意义上的数学论题——能够证明或不能证明，不如说是一个判断. 它表明，在一个可有限公理化的和谐理论中，可判定这个非正规的概念的形式化依赖于可表示这个概念.

定义 自然数上的关系 R 是*递归的*，当且仅当 R 在某个有限可公理化的和谐理论 (在含有 $\mathbf{0}$ 和 $\mathbf{S}$ 的语言中) 中可表示.

丘奇论题可以简洁地表达为，一个关系是可判定的当且仅当它是递归. 或者更精确地说，递归是可判定这个概念的准确表述. 这和我们在微积分中遇到的情况很类似. 直观上看，(定义在一个区间上的) 连续函数是一笔可以画成的图. 但在证明定理时，我们还是需要这个概念的形式表述. 因此，常用的 ε-δ 连续的概念就出现了. 有人会问，ε-δ 连续是不是直观上的连续的准确表述呢？实际上，ε-δ 连续的函数有很多，还包括处处不可微的函数，而这种函数的图像不是一笔可以画出的. 但不管精确与否，连续函数成为数学分析中很自然也很重要的一类函数. 207

在递归这个定义中，很多类似的情况也会发生. 有人会问：递归是可判定这个概念的准确表述吗？回答是相同的. 递归所定义的类 (递归关系) 太广了. 它还包括一些关系，这些关系的任何一个判定过程都将需要太长计算时间和太大的内存空间，以致于它们的判定不可能实现. 但不管怎么样，递归关系在数理逻辑中仍然是很自然也很重要的.

根据经验，递归关系的类中有不少于下面所列的成员.

(1) 迄今为止，数学家们所找到的可判定关系都是递归的.

(2) 一些研究者试图为理想化的计算设备给出精确的定义. 其中最著名的理想计算机是“图灵机”，它是由图灵 (Alan Turing) 在 1936 年提出的. (3.6 节中的寄存器就是由这个想法变化而来的.) 这个想法试图要设计出一种方法使得判定过程能够被有效地执行，但无论怎样，能够被这些计算机执行判定过程的关系恰好都是递归关系. (由于图灵分析在能行计算中发挥重要的作用，所以丘奇论题经常被称为丘奇　图灵论题.)

现在，递归关系有许多不同 (但等价) 的定义，这一事实也显示了这一概念的重要性和自然性.

本书中，我们在没有星号的定理中将不再使用可判定性这个不正规的定义，但在其他部分，我们仍然接受丘奇论题. 比如，当一个定理描述某个集合是非递归的时，我们就称这个集合是不可判定的.

任意在 Cn A_E 中可表示的关系显然是递归的，后面我们将证明反过来也是对的. 如果一个关系在任意一个可有限公理化的和谐理论中可表示，那么它在我们选定研究的理论中是可表示的. (当然，在我们的选择过程中，这是一个激发因素.)

“递归”这个词的使用是由历史上的意外事件造成的——甚至可以说是历史上的错误. 近来，一些数学家认为“可计算性”要比“递归”更接近概念的原意. 但在现在的文章中，“可计算性”用来指另一个我们即将给出的非正规概念. 对于关系，我们有可判定这一非正规的概念；对于函数，相似的概念即为可计算性. (作为简写符号，符号串 $a_1, \cdots, a_k$ 简记作 $\vec{a}$.)

208

***定义**　函数 $f: \mathbb{N}^k \to \mathbb{N}$ 是*可计算的*，当且仅当对于给定的 k 元自然数组 $\vec{a}$，存在一个能行的过程给出 $f(\vec{a})$ 的值.

例如，加法和乘法是可计算的. 在十进制情况下，计算加法和乘法的能行过程在小学就已经教过. (严格地说，在可计算性的概念中，我们研究的是*数字*，而不是数. 因为正是数字，像 317 或 XCI 这样的符号串，可以互现转换. 虽然如此，我们并不强调这一点.) 另一方面，从 $\mathbb{N}^k$ 到 $\mathbb{N}$ 的函数有不可数多个，其中只有可数多个是可计算的，因为只存在可数多个能行的过程.

和判定关系一样，我们希望给出可计算性这个非正规概念的数学化定义，我们将在下一个定理中讨论这个问题. 我们已经知道任意一个函数 $f: \mathbb{N}^k \to \mathbb{N}$ 也可看作 $\mathbb{N}$ 上的 $(k+1)$ 元关系：

$$\langle a_1, \cdots, a_k, b\rangle \in f \iff f(a_1, \cdots, a_k) = b.$$

我们曾经把函数和关系区别对待 (我们把关系称作函数的*图*)，但在当今的集合论中，两者是没有区别的. 当然，我们仍然可以用两种方式来看待函数.

***定理 33H**　设 $f: \mathbb{N}^k \to \mathbb{N}$ 是一个函数，则下列 3 个论述等价：

(a) f 是可计算的.

(b) 当将 f 看作一个关系时，f 是可判定的关系.

(c) 当将 f 看作一个关系时，f 是能行可枚举的关系.

证明　(a)⇒(b)：假设 f 是可计算的；我们要描述判定过程. 给定 $\langle a_1, \cdots, a_k, b\rangle$，首先计算 $f(a_1, \cdots, a_k)$，然后看这个结果是否等于 b. 如果是，f 是可判定的，否则就是不可判定的.

(b)⇒(c)：任意一个可判定的关系是能行可枚举的. 因为我们可以列出所有 $(k+1)$ 元数

组，在输出位置上列出符合关系的元.

(c)⇒(a)：假设我们已有 f(的图) 的能行枚举. 为了计算 $f(a_1, \cdots, a_k)$，我们逐个检验枚举出来的 $(k+1)$ 元组，直到我们找到以 $a_1, \cdots, a_k$ 为开始的那一组. 它的最后一个分量就是我们所要找的函数值. ■ 209

这样，根据丘奇论题我们可以说，f 是可计算的当且仅当 f(看作是关系) 是递归的. 即使我们不考虑递归函数类与不完全性定理的关系，它仍然是一类有趣的研究对象. 递归函数类代表了能被计算机程序所计算的函数类的上界. 如果我们不考虑计算时间和内存空间，递归函数就是能被计算机计算的函数.

现在我们来叙述这一节和下一节的计划. 我们的根本目的是为了得到 3.5 节中的定理. 但在证明这些定理之前，我们需要做一些准备工作；证明一些 (直观上可判定) 关系和 (直观上可计算) 函数在 Cn A_E 中是可表示的，当然，它们进而是递归的. 在这个过程中，我们将证明 (定理 34A) 在 CnA_E 中的递归等价于可表示. 本节还要证明一些与可表示性有关的基本事实，比如要证明，把有限数字序列转化为一个单一的数字的函数是可表示的. 在 3.4 节中，我们将把这些结果运用到一些特殊的关系和函数上，这些关系和函数与形式语言的语法特征有关.

作者清楚地知道，大家对 3.5 节中的定理的兴趣要比学习基础知识大得多. 如果大家相信，在 Cn A_E 中，直观上可判定的关系都是可表示的，并且直观上可计算的函数也是可函数表示的 (将要定义的概念)，那么，基础知识部分的证明，即使不是全部，大部分就变得没有必要了. 但我们希望大家仍要注意与这些结果相关的定义和结论.

3.3.4 按数字确定的公式

定理 33E 说明，对于在 Cn A_E 中可表示的关系，我们可以在 $\mathfrak{N}$ 中找到定义该关系的公式并且它可由 A_E 数字确定. 下面的定理为介绍数字确定打下基础.

定理 33I (a) 任意的原子公式是由 A_E 数字确定的.

(b) 如果 φ 和 ψ 由 A_E 数字确定，则 $\neg\varphi$ 和 $\varphi \to \psi$ 也由 A_E 数字确定.

(c) 如果 φ 由 A_E 数字确定，则下列的公式也由 A_E 数字确定 (通过对 φ 添加“有界量词”)，

210

$$\forall x(x < y \to \varphi),$$
$$\exists x(x < y \wedge \varphi).$$

证明 (a) 可从定理 33C 得到，(b) 是显然的，我们只需证明 (c). 我们考虑公式

$$\exists x(x < y \wedge \varphi(x, y, z))$$

其中只有变元 y 和 z 是自由的. 对于任意的自然数 a 和 b，我们要证明

$$A_E \vdash \exists x(x < \mathbf{S}^a\mathbf{0} \wedge \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}))$$

或

$$A_E \vdash \neg\exists x(x < \mathbf{S}^a\mathbf{0} \wedge \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0})).$$

情形 1：对于某个小于 a 的数 c,

$$A_E \vdash \varphi(\mathbf{S}^c\mathbf{0}, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}). \tag{1}$$

(这种情形出现当且仅当 $\exists x(x < \mathbf{S}^a\mathbf{0} \wedge \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}))$ 在 $\mathfrak{N}$ 中为真．) 还有

$$A_E \vdash \mathbf{S}^c\mathbf{0} < \mathbf{S}^a\mathbf{0}. \tag{2}$$

(1)、(2) 中的句子可以逻辑推出句子

$$\exists x(x < \mathbf{S}^a\mathbf{0} \wedge \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0})).$$

情形 2：对于每一个比 a 小的 c,

$$A_E \vdash \neg\, \varphi(\mathbf{S}^c\mathbf{0}, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}). \tag{3}$$

(这种情形出现当且仅当 $\forall x(x < \mathbf{S}^a\mathbf{0} \to \neg\, \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}))$ 在 $\mathfrak{N}$ 中为真．) 由引理 33A 可以知道

$$A_E \vdash \forall x(x < \mathbf{S}^a\mathbf{0} \to x = \mathbf{S}^0\mathbf{0} \vee \cdots \vee x = \mathbf{S}^{a-1}\mathbf{0}). \tag{4}$$

句子 (4) 和句子 (3) 一起 $(c = 0, \cdots, a-1)$ 可以逻辑推出

$$\forall x(x < \mathbf{S}^a\mathbf{0} \to \neg\, \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0})).$$

这等价于

$$\neg\, \exists x(x < \mathbf{S}^a\mathbf{0} \wedge \varphi(x, \mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0})).$$

这证明了 $\exists x(x < y \wedge \varphi(x, y, z))$ 是由 A_E 数字确定的．把这一结果用在 $\neg\, \varphi$ 上，我们就得到了对偶公式 $\forall x(x < y \to \varphi(x, y, z))$ 也是由 A_E 数字确定的． ■

211

情形 2 的讨论依赖于这个事实：x 受 $\mathbf{S}^a\mathbf{0}$ 的限制．我们将会看到 $\neg\, \psi(\mathbf{S}^0\mathbf{0}), \neg\, \psi(\mathbf{S}^1\mathbf{0}), \cdots$ 都有可能是 A_E 的推论，而 $\forall x \neg\, \psi(x)$ 不是 A_E 的推论．

在证明 Cn A_E 中许多关系可表示时，上面的定理是很有用的一个工具．例如，素数集可以用下面的公式来描述

$$\mathbf{S}^1\mathbf{0} < v_1 \wedge \forall x(x < v_1 \to \forall y(y < v_1 \to x \cdot y \neq v_1)).$$

这个公式定义了 $\mathfrak{N}$ 中的素数，并且根据上面的定理，它是由 A_E 数字确定的．因此，它可以表示 Cn A_E 中的素数集．

3.3.5　可表示函数

在很多情况下，使用函数要比用关系来得方便．设 $f : \mathbb{N}^m \to \mathbb{N}$ 是自然数上的 m 元函数，公式 φ 中只有 $v_1, \cdots, v_{m+1}$ 是自由变元．我们称 φ(在 Cn A_E 中) 函数表示 f，当且仅当对于 $\mathbb{N}$ 中的每一 m 元组 $a_1, \cdots, a_m$,

$$A_E \vdash \forall v_{m+1}[\varphi(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_m}\mathbf{0}, v_{m+1}) \leftrightarrow v_{m+1} = \mathbf{S}^{f(a_1, \cdots, a_m)}\mathbf{0}].$$

(这个句子中"←"方向等价于 $\varphi(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0},\mathbf{S}^{f(a_1,\cdots,a_m)}\mathbf{0})$，另一半"→"由唯一性的断言给出.)

定理 33J 若 φ 在 Cn A_E 中函数表示 f，则它在 Cn A_E 中表示 f(f 作为一个关系).

证明 不妨设 $m=1$. 由于 φ 函数表示 f，对于任意 b，有

$$A_E \vdash \varphi(\mathbf{S}^a\mathbf{0},\mathbf{S}^b\mathbf{0}) \leftrightarrow \mathbf{S}^b\mathbf{0}=\mathbf{S}^{f(a)}\mathbf{0}.$$

如果 $\langle a,b\rangle \in f$，即若 $f(a)=b$，则该等价号右侧是正确的，有

$$A_E \vdash \varphi(\mathbf{S}^a\mathbf{0},\mathbf{S}^b\mathbf{0}).$$

另一方面，等价号右侧却是被 A_E 拒绝的 (即，它的否定是可推出的)，因此

$$A_E \vdash \neg\,\varphi(\mathbf{S}^a\mathbf{0},\mathbf{S}^b\mathbf{0}).$$ ■

该定理的逆命题不成立，但我们可以修改一下这个公式使它成立.

定理 33K 设 f 是 $\mathbb{N}$ 上的函数，它 (作为关系) 在 Cn A_E 中是可表示的. 那么我们能找到在 Cn A_E 中函数表示 f 的公式 φ. 212

证明 为了简化符号，我们将 f 取成 $\mathbb{N}$ 上的一元函数. 考虑句子

$$\forall v_2[\varphi(\mathbf{S}^a\mathbf{0},v_2) \leftrightarrow v_2=\mathbf{S}^{f(a)}\mathbf{0}],$$

它等价于以下两个句子的合取：

$$\varphi(\mathbf{S}^a\mathbf{0},\mathbf{S}^{f(a)}\mathbf{0}) \tag{1}$$

和

$$\forall v_2[\varphi(\mathbf{S}^a\mathbf{0},v_2) \to v_2=\mathbf{S}^{f(a)}\mathbf{0}]. \tag{2}$$

如果 φ 能表示 f，则句子 (1) 是 A_E 的一个定理. 句子 (2) 是对唯一性的断言，我们必须构造出公式 φ，使得 (2) 也是 A_E 的一个定理.

以公式 θ 表示 f(看作二元关系) 为开始，令 φ 为

$$\theta(v_1,v_2)\wedge\forall z(z<v_2\to\neg\,\theta(v_1,z)).$$

我们可以把公式 (2) 改写成

$$\forall v_2[\theta(\mathbf{S}^a\mathbf{0},v_2)\wedge\forall z(z<v_2\to\neg\,\theta(\mathbf{S}^a\mathbf{0},z))\to v_2=\mathbf{S}^{f(a)}\mathbf{0}]. \tag{2'}$$

为了证明这个公式是 A_E 的定理，只需要证明

$$A_E\cup\{\theta(\mathbf{S}^a\mathbf{0},v_2),\forall z(z<v_2\to\neg\,\theta(\mathbf{S}^a\mathbf{0},z))\}\vdash v_2=\mathbf{S}^{f(a)}\mathbf{0}.$$

我们把这个假设集合 ("$\vdash$" 左侧部分) 记为 Γ. 由于 L3$\in A_E$，因此只需要证明

$$\Gamma\vdash v_2\not<\mathbf{S}^{f(a)}\mathbf{0} \tag{3}$$

和

$$\Gamma \vdash \mathbf{S}^{f(a)}\mathbf{0} \nless v_2. \tag{4}$$

式 (4) 较容易得到，因为从 Γ 的后一部分我们能够得到

$$\mathbf{S}^{f(a)}\mathbf{0} < v_2 \to \neg\, \theta(\mathbf{S}^a\mathbf{0}, \mathbf{S}^{f(a)}\mathbf{0})$$

并且，我们知道

$$A_E \vdash \theta(\mathbf{S}^a\mathbf{0}, \mathbf{S}^{f(a)}\mathbf{0}). \tag{5}$$

为了得到式 (3)，首先我们注意到 A_E 的下列定理：

$$v_2 < \mathbf{S}^{f(a)}\mathbf{0} \leftrightarrow v_2 = \mathbf{S}^0\mathbf{0} \vee \cdots \vee v_2 = \mathbf{S}^{f(a)-1}\mathbf{0} \tag{6}$$

和

$$\neg\, \theta(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}) \ \ , b = 0, \cdots, f(a) - 1. \tag{7}$$

公式 (6) 和公式 (7) 蕴涵公式

$$v_2 < \mathbf{S}^{f(a)}\mathbf{0} \to \neg\, \theta(\mathbf{S}^a\mathbf{0}, v_2). \tag{8}$$

213 由于 $\theta(\mathbf{S}^a\mathbf{0}, v_2) \in \Gamma$，所以我们可以得到公式 (3).

这证明了公式 (2) 是 A_E 的一个定理；公式 (5) 和公式 (8) 同样证明了公式 (1) 是 A_E 的定理. ■

接下来，我们要证明一些基本函数 (在 Cn A_E 中) 是可表示的，并且可表示函数具有某些封闭的性质. 在本章的后续部分中，当我们提到函数或关系是可表示的时，是指它们在理论 Cn A_E 中可表示，“在 Cn A_E 中” 通常省略不说.

我们看一个简单的例子，一个 m 元函数可以用方程

$$v_{m+1} = t.$$

来表示. 实际上，当 t 中的变元在 $v_1, \cdots, v_m$ 中时，任意一个这样的方程都定义了一个 $\mathfrak{N}$ 中的 m 元函数 f. (f 在 $\langle a_1, \cdots, a_m\rangle$ 上的值是，即当 v_i 取为 a_i 时，t 在 $\mathfrak{N}$ 中所对应的值，$1 \leqslant i \leqslant m$.) 而且，我们知道任意方程是由 A_E 数字确定的，因此 f 作为关系可以由方程表示. 事实上，方程甚至可以函数表示 f，因为句子

$$\forall v_{m+1}[v_{m+1} = t(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_m}\mathbf{0}) \leftrightarrow v_{m+1} = \mathbf{S}^{f(a_1, \cdots, a_m)}\mathbf{0}]$$

逻辑等价于

$$t(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_m}\mathbf{0}) = \mathbf{S}^{f(a_1, \cdots, a_m)}\mathbf{0},$$

它在 $\mathfrak{N}$ 中是真的无量词句子. (这里 $t(u_1, \cdots, u_m)$ 是将 v_1 换成 u_1, v_2 换成 $u_2, \cdots\cdots$ 后得到的项.) 例如：

(1) 后继函数可由下面的方程 (函数) 表示：

$$v_2 = \mathbf{S}v_1.$$

(2) 任意的常值函数是可表示的. m 元常值函数 (函数值为 b) 可以用下面的方程表示：

$$v_{m+1} = \mathbf{S}^b\mathbf{0}.$$

(3) 投影函数 $(1 \leqslant i \leqslant m)$

$$I_i^m(a_1, \cdots, a_m) = a_i$$

可以表示为

$$v_{m+1} = v_i.$$

(4) 加法、乘法和幂乘运算分别由下列方程表示：

$$v_3 = v_1 + v_2,$$

$$v_3 = v_1 \cdot v_2,$$

$$v_3 = v_1\mathbf{E}v_2,$$

214

但是，大家不要被这些简单的例子误导了，并不是每一个可表示的函数都可以用方程表示.

我们下面要证明，可表示函数族在复合运算下是封闭的. 为了简化书写，我们只考虑 $\mathbb{N}$ 上的一元函数 f，此处

$$f(a) = g(h_1(a), h_2(a)).$$

假设 g 由 ψ 函数表示，h_i 由 θ_i 函数表示. 为了表示 f，考虑

$$\forall y_1 \forall y_2(\theta_1(v_1, y_1) \to \theta_2(v_1, y_2) \to \psi(y_1, y_2, v_2))$$

和

$$\exists y_1 \exists y_2(\theta_1(v_1, y_1) \land \theta_2(v_1, y_2) \land \psi(y_1, y_2, v_2)).$$

(由于 $\psi(y_1, y_2, v_2)$ 等同于 $g(y_1, y_2) = v_2$，$\theta_i(v_1, y_1)$ 等同于 $h_i(v_1) = y_i$. 那么第一个公式表示，“对于任意的 y_1, y_2，若 $h_1(v_1) = y_1$，$h_2(v_1) = y_2$，则 $g(y_1, y_2) = v_2$. ” 第二公式表示，“存在 y_1, y_2，使得 $h_1(v_1) = y_1$，$h_2(v_1) = y_2$ 和 $g(y_1, y_2) = v_2$. ” 这二者都表示 “$g(h_1(v_1), h_2(v_1)) = v_2$”. 现在有两个选择，因为表达式唯一时，两个量词都可以使用.)

实际上，两个公式都可行；令 φ 为公式

$$\forall y_1 \forall y_2(\theta_1(v_1, y_1) \to \theta_2(v_1, y_2) \to \psi(y_1, y_2, v_2)).$$

考虑任意的自然数 a，有

$$\forall v_2[\psi(\mathbf{S}^{h_1(a)}\mathbf{0}, \mathbf{S}^{h_2(a)}\mathbf{0}, v_2) \leftrightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}]. \tag{1}$$

$$\forall y_1[\theta_1(\mathbf{S}^a\mathbf{0}, y_1) \leftrightarrow y_1 = \mathbf{S}^{h_1(a)}\mathbf{0}]. \tag{2}$$

$$\forall y_2[\theta_2(\mathbf{S}^a\mathbf{0}, y_2) \leftrightarrow y_2 = \mathbf{S}^{h_2(a)}\mathbf{0}]. \tag{3}$$

要证明

$$\forall v_2(\varphi(\mathbf{S}^a\mathbf{0}, v_2) \leftrightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}), \tag{4}$$

即

$$\forall v_2(\forall y_1 \forall y_2[\theta_1(\mathbf{S}^a\mathbf{0}, y_1) \to \theta_2(\mathbf{S}^a\mathbf{0}, y_2) \to \psi(y_1, y_2, v_2)] \leftrightarrow v_2 = \mathbf{S}^{f(a)}\mathbf{0}). \tag{5}$$

由式 (1), (2), (3) 可以推出式 (4), 见习题 4.

对于更一般的情况，有：

定理 33L 设 g 是 n 元函数，$h_1, \cdots, h_n$ 是 m 元函数，f 由下面的公式定义：

215

$$f(a_1, \cdots, a_m) = g(h_1(a_1, \cdots, a_m), \cdots, h_n(a_1, \cdots, a_m)).$$

如果 $g, h_1, \cdots, h_n$ 是可函数表示的，那么我们可以找到一个公式函数表示 f.

上一个定理证明了 $m = 1, n = 2$ 的情况. 但一般情况的证明也是用同样的方法. 为了得到如下的一个函数：

$$f(a, b) = g(h(a), b),$$

注意到

$$f(a, b) = g(h(I_1^2(a, b)), I_2^2(a, b)).$$

反复利用上述定理 (两次)，可以证明 f 是可表示的 (如果 g 和 h 是可表示的).

为了简化对任意多个变元的函数的讨论，我们使用向量符号. 例如，上述定理中的方程可以写成

$$f(\vec{a}) = g(h_1(\vec{a}), \cdots, h_n(\vec{a})).$$

另一个重要的闭性质是：Cn A_E 中的函数可表示性在 “最小零” 运算下是封闭的.

定理 33M 假设 $(m+1)$ 元函数 g 是可表示的，并且对于每一组 $a_1, \cdots, a_m$，存在 b 使得

$$g(a_1, \cdots, a_m, b) = 0.$$

那么能找到一个公式，它能表示 m 元函数 f，其中

$$f(a_1, \cdots, a_m) = \text{最小的}b, \text{使得}g(a_1, \cdots, a_m, b) = 0.$$

(我们可以用向量符号简化上述方程：

$$f(\vec{a}) = \text{最小的}b, \text{使得}g(\vec{a}, b) = 0.$$

这个运算称为最小零运算，用下面的符号表示：

$$f(\vec{a}) = \mu b[g(\vec{a}, b) = 0]$$

且该运算符称作 “μ 运算符”.)

证明 为了简化书写，我们只取 $m=1$；这样 $f(a)=b$ 当且仅当 $g(a,b)=0$ 并且对于所有的 $c<b$，$g(a,c)\neq 0$.

如果 ψ 表示 g，我们只需要将上述等式的右边形式化，就可以得到表示 f(看作关系) 的公式：

$$\psi(v_1,v_2,\mathbf{0})\wedge\forall y(y<v_2\to\neg\,\psi(v_1,y,\mathbf{0})).$$

216

这个公式定义了 f(的图像)，并且它是由 A_E 数字确定的. ■

3.3.6 编目

现在我们给出 (在 Cn A_E 中) 可表示的函数和关系的编目，特别还包括编码和解码序列函数.

(0) 从定理 33I 可以得到，任何 (在 $\mathfrak{N}$ 中) 由无量词公式定义的关系是可表示的. 并且可表示关系的类在交、并和余运算下是封闭的. 如果 R 是可表示的，则

$$\{\langle a_1,\cdots,a_m,b\rangle|\text{对于所有的 } c<b,\langle a_1,\cdots,a_m,c\rangle\in R\}$$

和

$$\{\langle a_1,\cdots,a_m,b\rangle|\text{对于某个 } c<b,\langle a_1,\cdots,a_m,c\rangle\in R\}$$

也是可表示的.

例如，任意的有限关系都有一个无量词公式的定义，同样序关系也是一样.

(1) 关系 R 是可表示的当且仅当它的特征函数 K_R 是可表示的. (当 $\vec{a}\in R$ 时，则函数 $K_R(\vec{a})=1$，否则 $K_R(\vec{a})=0$.)

证明 ($\Leftarrow$) 设 R 为一元关系 ($\mathbb{N}$ 的子集)，它的特征函数 K_R 由 $\psi(v_1,v_2)$ 表示. 我们断言 R 可以由 $\psi(v_1,\mathbf{S0})$ 表示. 因为它定义了 R，并且是由 A_E 数字确定的.

($\Rightarrow$) 设公式 $\varphi(v_1)$ 表示 R. 那么根据上一段中的相同原因

$$(\varphi(v_1)\wedge v_2=\mathbf{S0})\vee(\neg\,\varphi(v_1)\wedge v_2=\mathbf{0})$$

表示 K_R(的图像). (实际上，大家可以证明，这个公式函数表示 K_R.) ■

(2) 如果 R 是一个可表示的二元关系，并且 f 和 g 是可表示的函数，则

$$\{\vec{a}|\langle f(\vec{a}),g(\vec{a})\rangle\in R\}$$

是可表示的. 对于 m 元的关系 R 和函数 $f_1,\cdots,f_m$，结论也是一样的.

证明 R 的特征函数在 $\vec{a}$ 的值为 $K_R(f(\vec{a}),g(\vec{a}))$，这样可表示函数通过复合就能得到这一结果. ■

例如，如果我们假设 R 是可表示的三元关系，则集合

$$\{\langle x,y\rangle|\langle y,x,x\rangle\in R\}$$

217

可表示为

$$\{\langle x,y\rangle|\langle I_2^2(x,y),I_1^2(x,y),I_1^2(x,y)\rangle\in R\}.$$

我们可以通过重新排列变元和重复变元的方法描述可表示关系.

(3) 如果 R 是可表示的二元关系，则

$$P = \{\langle a,b\rangle | \text{对于某个} c \leqslant b, \text{使得} \langle a,c\rangle \in R\}.$$

也是可表示的二元关系.

证明　我们从编目 0 可知，如果

$$Q = \{\langle a,b\rangle | \text{对于某个} c < b, \text{使得} \langle a,c\rangle \in R\},$$

则 Q 是可表示的. 并且

$$\begin{aligned}\langle a,b\rangle \in P \quad &\Leftrightarrow \quad \langle a, S(b)\rangle \in Q \\ &\Leftrightarrow \quad \left\langle I_1^2(a,b), S(I_2^2(a,b))\right\rangle \in Q.\end{aligned}$$

再根据编目 2，我们可以得出 P 是可表示的. ■

更一般地说，如果 R 是一个可表示的 $(m+1)$ 元关系，则关系

$$\{\langle a_1,\cdots,a_m,b\rangle | \text{对于某个} c \leqslant b, \text{使得} \langle a_1,\cdots,a_m,c\rangle \in R\}$$

也是可表示的. 这个关系可以用向量表示为

$$\{\langle \vec{a},b\rangle | \text{对于某个} c \leqslant b, \text{使得} \langle \vec{a},c\rangle \in R\}.$$

类似地，

$$\{\langle \vec{a},b\rangle | \text{对于所有} c \leqslant b, \text{使得} \langle \vec{a},c\rangle \in R\}$$

也是可表示的.

(4) 整除关系

$$\{\langle a,b\rangle | a \text{ 整除 } b, a, b \text{ 属于 } \mathbb{N}\}$$

是可表示的.

证明　我们知道，a 整除 b 当且仅当对于某个 $q \leqslant b$，使得 $a \cdot q = b$. 又由于 $\{\langle a,b,q\rangle | a\cdot q = b\}$ 是由无量词公式定义的，所以它是可表示的. 运用上面几个结论，我们可以得到整除关系. (更具体地说，从编目 3，我们可以得到关系

$$R = \{\langle a,b,c\rangle | \text{对于某个} q \leqslant c, \text{使得} a \cdot q = b\}$$

的可表示性，并且我们还能得到 a 整除 b 当且仅当 $\langle a,b,b\rangle \in R$.) ■

(5) 素数集合是可表示的.

(6) 相邻素数对的集合是可表示的.

218

证明　$\langle a,b\rangle$ 是相邻的素数对当且仅当 a, b 均为素数，$a < b$ 并且不存在素数 c，使得 $a < c < b$. 这个等价条件右边很容易由数字确定的公式公式化. ■

请注意，到现在为止，我们还没有用到幂乘运算是可表示的这一事实. 但在 3.8 节中，我们会使用这些结果.

在整个编目过程中，我们一直在有效地构建一种“语言”$\mathcal{L}$，使得 (在 $\mathfrak{N}$ 中)$\mathcal{L}$ 可定义的一切 (任意的关系和函数) 在我们的理论中都是可表示的. 定理 33I 说明了 (a) $\mathcal{L}$ 中有原子公式，(b) 所有的命题联结词都可以使用，(c) 有界量词在 $\mathcal{L}$ 中允许使用. (在一般情况下，无界量词是不允许使用的.) 然后，我们的编目逐渐地增加了一些特殊的谓词符号和函数符号；编目 6 为“相邻的素数关系”增加了一个二元谓词符号；编目 7 将为素数枚举函数增加一个函数符号. 定理 33L 证明了在 $\mathcal{L}$ 的表达式中使用这些函数符号是可行的.

(7) 设一个函数在 a 的值是第 $(a+1)$ 个素数 p_a，则该函数是可表示的 (即 $p_0=2, p_1=3, p_2=5, p_3=7, p_4=11$, 等等).

证明 我们知道，$p_a=b$ 当且仅当 b 是素数并且存在某个 $c \leqslant b^{a^2}$，使得下列条件 (i)~(iii) 成立.

(i) 2 不能整除 c.

(ii) 对于任意的 $q<b$ 和 $r \leqslant b$，如果 $\langle q,r\rangle$ 是相邻的素数对，则对于所有的 $j<c$，有

$$q^j \text{ 整除 } c \Longleftrightarrow r^{j+1} \text{ 整除 } c.$$

(iii) b^a 整除 c 但 b^{a+1} 并不整除 c.

虽然这个等价式并不显然，但至少我们知道该等价式的右半部分是可表示的. 为了验证这个等价式，我们先设 $p_a=b$，那么可以取

$$c=2^0\cdot 3^1\cdot 5^2\cdot\cdots\cdot p_a^a.$$

容易验证 c 满足所有的上述 3 个条件. 反之，假设 c 是满足条件 (i)~(iii) 的数. 我们断言，c 一定等于

$$2^0\cdot 3^1\cdot\cdots\cdot b^a\cdot \text{更大的一些素数的幂}.$$

显然，由 (i) 知，c 中 2 的指数为 0. 我们再利用 (ii)，就可以求出素数 b. 但由 (iii) 知，b 的指数是 a，因此，b 一定是第 $(a+1)$ 个素数 p_a. ■ 219

当我们把数的有限序列编码成单独的一个数字时，这个函数将起到非常重要的作用. 现在设

$$\begin{aligned}\langle a_0,\cdots,a_m\rangle &= p_0^{a_0+1}\cdots p_m^{a_m+1}\\ &= \prod_{i\leqslant m} p_i^{a_i+1}\end{aligned}$$

当 $m=-1$ 时这个公式也成立，我们定义 $\langle\ \rangle$=1. 例如，

$$\langle 2,1\rangle = 2^3\cdot 3^2=72.$$

这个式子表示 $\langle 2,1\rangle$ 可以编码转化为 72.

我们还能有其他的途径转化数对和有限长的数字序列. 在 3.8 节中，我们将使用配对函数

$$J(a,b)=\frac{1}{2}[(a+b)^2+3a+b]$$

这个函数的优点是它以多项式的速度增长，比 $2^{a+1}3^{b+1}$ 的增长速度慢. 这里还有一种不同的转化方法，比如，如果我们要转化 24, 117, 11(以那个序). 首先我们把它们转化成 9 进制

的数字:26, 140, 12. 然后将这些数连起来，中间用 9 来间隔：269140912. 这样，这个三元组就被转化成 (10 进制)269,140,912. 这个方法看起来需要技巧，但得到的结果要比 $2^{25}3^{118}5^{12}$ 小得多，$2^{25}3^{118}5^{12}$ 在 10 进制中是一个 73 位的数.

(8) 对于每个 m，在 $a_0, \cdots, a_m$ 的值为 $\langle a_0, \cdots, a_m\rangle$ 的函数是可表示的.

(9) 存在一个可表示函数 (它在 $\langle a, b\rangle$ 的值记作 $(a)_b$) 满足对于 $b \leqslant m$，有

$$(\langle a_0, \cdots, a_m\rangle)_b = a_b.$$

(我们称之为"解码"函数. 例如，$(72)_0 = 2$, $(72)_1 = 1$.)

证明　定义 $(a)_b = n$，n 为使得 $a = 0$ 或 p_b^{n+2} 不整除 a 的元素中最小的一个. (这样的 n 总是存在的.) 可以知道，$(0)_b = 0$，对于 $a \neq 0$，$(a)_b$ 比 a(但不能小于 0) 的素数分解中的 p_b 的指数小 1. 这样对于 $b \leqslant m$，

$$(\langle a_0, \cdots, a_m\rangle)_b = a_b.$$

为了证明该函数的可表示性，我们使用最小零运算符. 令

$$R = \{\langle a, b, n\rangle | a = 0 \text{ 或 } p_b^{n+2} \text{ 不整除 } a\}.$$

则 $(a)_b = \mu n[K_{\overline{R}}(a, b, n) = 0]$，其中 $\overline{R}$ 是 R 的补. ■

220

由于上面证明中所用的方法非常有用，所以我们单独把它列出来：

定理 33N　设 R 是一个可表示的关系，它使得对于任意 $\vec{a}$，都存在某个 n 使得 $\langle \vec{a}, n\rangle \in R$. 那么下面的函数是可表示的：

$$f(\vec{a}) = n, n\text{为使得 } \langle \vec{a}, n\rangle \text{ 最小的元素}.$$

证明　$f(\vec{a}) = \mu n[K_{\overline{R}}\langle \vec{a}, n\rangle = 0]$. ■

我们以后将这个式子简写为 $f(\vec{a}) = \mu n[\langle \vec{a}, n\rangle \in R]$.

(10) 我们称 b 是一个 *数字序列* 当且仅当对某个 $m \geqslant -1$ 和 $a_0, \cdots, a_m$，使得

$$b = \langle a_0, \cdots, a_m\rangle.$$

(当 $m = -1$ 时，我们取 $\langle\,\rangle = 1$.) 那么数字序列的集合是可表示的.

证明　习题 5. ■

(11) 存在一个可表示函数 lh，使得

$$\text{lh}\langle a_0, \cdots, a_m\rangle = m + 1.$$

(此处"lh"代表"长度"例如，lh72=2.)

证明　我们只需要定义 lh $a = n$，n 为使得 $a = 0$ 或 p_n 不整除 a 的元素中最小的一个就够了. ■

(12) 存在一个可表示函数 (它在 $\langle a, b\rangle$ 上的取值称为 a 在 b 上的 *约束*，记作 $a \upharpoonright b$) 使得，对于任意的 $b \leqslant m + 1$，有

$$\langle a_0, \cdots, a_m \rangle \upharpoonright b = \langle a_0, \cdots, a_{b-1} \rangle.$$

证明 令 $a \upharpoonright b =$ 最小的 n，使得 $a = 0$ 或者 $n \neq 0$，同时对于任意的 $j < b$，$k < a$，

$$p_j^k \text{ 整除 } a \Rightarrow p_j^k \text{ 整除 } n. \qquad \blacksquare$$

(13) (原始递归式) 对于一个 $(k+1)$ 元函数 f，我们可以找到另一个函数 $\bar{f}$，使得对于所有的 $j < a$，$\bar{f}(a, b_1, \cdots, b_k)$ 能够编码 $f(j, b_1, \cdots, b_k)$ 的值. 特别地，令

$$\bar{f}(a, \vec{b}) = \langle f(0, \vec{b}), \cdots, f(a-1, \vec{b}) \rangle.$$

例如，$\bar{f}(0, \vec{b}) = \langle\ \rangle = 1$ 编码了函数 f 的第一个 0 值. $\bar{f}(1, \vec{b}) = \langle f(0, \vec{b}) \rangle$. 无论哪种情况，$\bar{f}(a, \vec{b})$ 都是长度为 a 的数字序列，它编码了函数 f 的第一个 a 值.

221

现在假设给定一个 $(k+2)$ 元函数 g，则存在唯一的函数 f 使得

$$f(a, \vec{b}) = g(\bar{f}(a, \vec{b}), a, \vec{b}).$$

例如，

$$f(0, \vec{b}) = g(\langle\ \rangle, 0, \vec{b}),$$
$$f(1, \vec{b}) = g(\langle f(0, \vec{b}) \rangle, 1, \vec{b}).$$

(从直观上看，函数 f 的存在性和唯一性是显然的. 如果要证明，我们可以用 1.4 节中的递归定理先得到 $\bar{f}$，进而得到 f.)

定理 33P 设 g 是 $(k+2)$ 元函数，f 是唯一的一个 $(k+1)$ 元函数，使得对于所有的 a 和 $\vec{b}$(k 元)，

$$f(a, \vec{b}) = g(\bar{f}(a, \vec{b}), a, \vec{b}).$$

如果 g 是可表示的，则 f 也是可表示的.

证明 首先我们断言，$\bar{f}$ 是可表示的.

这个断言可以从下面的事实中得到.

$\bar{f}(a, \vec{b}) = s$，s 是长度为 a 的数字序列并且

对于 $i < a$，$(s)_i = g(s \upharpoonright i, i, \vec{b})$ 的最小者.

因为 $f(a, \vec{b}) = g(\bar{f}(a, \vec{b}), a, \vec{b})$，并且右边的函数是可表示的，所以 f 是可表示的. $\blacksquare$

实际上，"原始递归" 这个词更多地应用于另一种更简单的形式，见习题 8.

(14) 设 F 是可表示的函数，则在 $a, \vec{b}$ 上的取值为

$$\prod_{i<a} F(i, \vec{b})$$

的函数也是可表示的. 如果用 Σ 代替 Π，得到的函数同样也是可表示的. (对于 $a = 0$，我们使用标准的约定：空的乘积 (即没有数相乘) 为 1，空的和为 0.)

证明 我们把所定义的函数称为 G，则

$$G(0, \vec{b}) = 1,$$
$$G(a+1, \vec{b}) = F(a, \vec{b}) \cdot G(a, \vec{b}).$$

再运用习题 8 的结果即可证明. ■

(15) 定义 a,b 的连接函数 $a*b$ 为

222

$$a*b=a\cdot\prod_{i<\mathrm{lh}b}p_{i+\mathrm{lh}a}^{(b)_i+1}.$$

这是 a,b 的可表示函数, 并且

$$\langle a_1,\cdots,a_m\rangle*\langle b_1,\cdots,b_n\rangle=\langle a_1,\cdots,a_m,b_1,\cdots,b_n\rangle.$$

连接运算符在连接数字序列的运算中还有另外一些性质.

(16) 我们还需要定义"大星号"运算符. 令

$$*_{i<a}f(i)=f(0)*f(1)*\cdots*f(a-1).$$

对于一个可表示函数 F, 在 $a,\vec{b}$ 上取值为 $*_{i<a}F(i,\vec{b})$ 的函数是可表示的.

证明

$$*_{i<0}F(i,\vec{b})=\langle\ \rangle=1\ 和$$
$$*_{i<a+1}F(i,\vec{b})=*_{i<a}F(i,\vec{b})*F(a,\vec{b}).$$

这和编目 14 很相似. ■

习题

1. 证明: 在结构 $(\mathbb{N};\cdot,E)$ 中, 我们能够定义加法关系 $\{\langle m,n,m+n\rangle|m,n\in\mathbb{N}\}$. 进一步证明在结构 {0} 中, 序关系 $<$, 后继关系 $\{\langle n,S(n)\rangle|n\in\mathbb{N}\}$ 都是可定义的. (说明: 如果把结构 $(\mathbb{N};\cdot,E)$ 简化为 $(\mathbb{N};E)$, 这个结果还能加强. 在此处, 乘法关系可以通过指数运算的一个法则: $(d^a)^b=d^{ab}$ 来定义.)
2. 证明定理 33C, 即 (在 $\mathfrak{N}$ 中) 真的无量词句子都是 A_E 的定理.
3. 我们称一个理论 T(在含 **0** 和 **S** 的语言中) 是 ω完全的, 当且仅当对于任何公式 φ 和任意变元 x, 如果对任意自然数 n, $\varphi^x_{\mathbf{S}^n\mathbf{0}}$ 都属于 T, 则 $\forall x\varphi$ 属于 T. 证明: 若 T 在 $\mathfrak{N}$ 的语言中是 ω 完全的和谐理论并且如果 $A_E\subseteq T$, 则 $T=\mathrm{Th}\,\mathfrak{N}$.
4. 证明: 在定理 33L 的证明过程中, 公式 (4) 可以由 (1)、(2)、(3) 逻辑推出.
5. 证明: 序列数字的集合是可表示的 (编目 10).
6. 3 是序列数字吗? lh3 等于多少? 求出 $(1*3)*6$ 和 $1*(3*6)$.
7. 验证下列事实:
   (a) $a+1<p_a$.
   (b) $(b)_k\leqslant b$; 当且仅当 $b=0$ 时等号成立
   (c) $\mathrm{lh}a\leqslant a$; 当且仅当 $a=0$ 时等号成立.
   (d) $a\upharpoonright i\leqslant a$.
   223
   (e) $\mathrm{lh}(a\upharpoonright i)$ 等于 i 和 $\mathrm{lh}a$ 中的较小者.
8. 设 g 和 h 是可表示函数, 同时假设

$$f(0,b)=g(b),$$
$$f(a+1,b)=h(f(a,b),a,b).$$

　证明 f 是可表示的.

9. 证明：存在一个可表示函数 f 使得对于任意的 $n, a_0, \cdots, a_n$,

$$f(\langle a_0, \cdots, a_n \rangle) \quad = \quad a_n.$$

(例如，$f(72)=1$，$f(750)=2$.)

10. 设 R 是可表示的关系，g 和 h 是可表示的函数. 证明：f 是可表示的，其中

$$f(\vec{a}) = \begin{cases} g(\vec{a}) & \text{若 } \vec{a} \in R, \\ h(\vec{a}) & \text{若 } \vec{a} \notin R. \end{cases}$$

11. (单调递归) 设 R 为 $\mathbb{N}$ 上可表示的二元关系. 令 C 为 $\mathbb{N}$ 最小的子集 (即所有子集的交集) 使得对于所有的 $n, a_0, \cdots, a_{n-1}, b$,

$$\langle\langle a_0, \cdots, a_{n-1} \rangle, b\rangle \in R \quad \text{且} \quad a_i \in C \quad (\text{对于所有的 } i < n) \Rightarrow b \in C.$$

进一步假设 (1) 对于所有的 $n, a_0, \cdots, a_{n-1}, b$,

$$\langle\langle a_0, \cdots, a_{n-1} \rangle, b\rangle \in R \Rightarrow a_i < b \quad (\text{对于所有的 } i < n),$$

和 (2) 存在一个可表示的函数 f 使得对于所有 $n, a_0, \cdots, a_{n-1}, b$,

$$\langle\langle a_0, \cdots, a_{n-1} \rangle, b\rangle \in R \Rightarrow n < f(b)$$

证明：C 是可表示的. (从某种意义上说，C 是由 R 生成的. 一般地，$C \neq \varnothing$ 是因为如果 $\langle\langle\ \rangle, b\rangle \in R$，则 $b \in C$.)

3.4 语法的算术化

在这一节中，我们主要讨论两个主题：

(1) 一些关于合式公式的断言 (通过把数指派给表达式) 可以转化成关于自然数的断言.

(2) 这些关于自然数的断言 (自然语言表述) 在很多情况下可以转化成形式语言. 这样得到的许多结论可以通过理论 Cn A_E 得到证明.

由关于数字的表达式，我们能够构造出关于公式的间接表达式 (甚至关于公式自身的表达式！). 在 3.5 节中，我们将用这种方法得到关于不可定义性和不可判定性的一些结果. [224]

哥德尔数

首先，我们要把数字指派给形式语言中的表达式，我们语言中所用到的符号都在表 3-1 中列出.

表 3-1

参数	逻辑符号
0. $\forall$	1. (
2. **0**	3.)
4. **S**	5. $\neg$
6. $<$	7. $\to$
8. $+$	9. $=$
10. $\cdot$	11. v_1
12. **E**	13. v_2

存在一个函数 h, 它把每个符号左边的整数指派给这个符号. 这样 $h(\forall)=0, h(\mathbf{0})=2, h(v_i)=9+2i$. 为了使这个数列能有更广泛的应用, 我们要假设含有 $\mathbf{0}$ 和 $\mathbf{S}$ 的这些语言中的元素是 *递归编号* 的. 根据这个假设, 我们可以得到一个函数 h 将语言的参数集一对一映射到偶数内, 并且确保以下两个关系在 $\mathrm{Cn}\ A_E$ 中都是可表示的:

$$\{\langle k,m\rangle \mid k \text{ 是 } h \text{ 在某个 } m \text{ 元谓词符号上的取值}\}$$

和

$$\{\langle k,m\rangle \mid k \text{ 是 } h \text{ 在某个 } m \text{ 元函数符号上的取值}\}.$$

当然, 在 $\mathfrak{N}$ 的语言中, 这两个集合都是有限的. 第一个集合为 $\{\langle 6,2\rangle\}$, 第二个为

$$\{\langle 2,0\rangle,\langle 4,1\rangle,\langle 8,2\rangle,\langle 10,2\rangle,\langle 12,2\rangle\}.$$

h 在逻辑符号上的值仍然按照前面的定义; 这样 $h(s)$ 是与每个逻辑符号 s 对应的奇数.

对于语言的表达式 $\varepsilon=s_0\cdots s_n$, 我们定义它的哥德尔数 $\sharp(\varepsilon)$ 为

$$\sharp(s_0\cdots s_n)=\langle h(s_0),\cdots,h(s_n)\rangle.$$

比如, 对 $\mathfrak{N}$ 的语言使用上述的函数 h, 我们可以得到

$$\begin{aligned}\sharp(\exists v_3 v_3=\mathbf{0})&=\sharp((\neg\ \forall v_3(\neg\ = v_3\mathbf{0})))\\&=\langle 1,5,0,15,1,5,9,15,2,3,3\rangle\\&=2^2\cdot 3^6\cdot 5^1\cdot 7^{16}\cdot 11^2\cdot 13^6\cdot 17^{10}\cdot 19^{16}\cdot 23^3\cdot 29^4\cdot 31^4.\end{aligned}$$

225

这是一个很大的数, 是 1.3×10^{75} 的序数, 对于一个表达式的集合 Φ, 我们定义这个集合的哥德尔数为

$$\sharp\Phi=\{\sharp(\varepsilon)|\varepsilon\in\Phi\}$$

对于表达式序列 $\langle\alpha_0,\cdots,\alpha_n\rangle$(例如一个推理), 我们做如下指派

$$\mathcal{G}(\langle\alpha_0,\cdots,\alpha_n\rangle)=\langle\sharp\alpha_0,\cdots,\sharp\alpha_n\rangle.$$

我们现在要证明与与哥德尔数有关的各种关系和函数在 $\mathrm{Cn}\ A_E$ 中都是可表示的 (进而是递归的). 与在前一节中一样, 当我们谈及一个关系或函数是可表示的时 (没有指定一个理论) 是指它在理论 $\mathrm{Cn}\ A_E$ 中是可表示的.

我们将对所使用的语言 (自然语言, 尽管与我们通常所认为的自然语言有很大的不同.) 进行一些缩写, 对于 "存在一个数 a", 我们写作 "$\exists a$" . 那么, "$\exists a,b<c$" 表示 "存在数 a 和 b, 它们都比 c 小." 类似地, 我们还要使用 "$\forall$" . 我们在第 2 章中没有使用这些缩写是因为我们担心读者会对形式语言和元语言 (自然语言) 产生混淆, 但现在我们相信读者已经能够避免产生这样的误解了.

(1) 变元的哥德尔数集合是可表示的.

证明 这个集合为 $\{a|(\exists b<a)a=\langle 11+2b\rangle\}$. 我们从上一节的结果知道, 这个集合是可表示的. ■

(2) 项的哥德尔数集合是可表示的.

证明 项的概念是通过归纳的方法定义的, 因此项是由较小的哥德尔数组组成的. 由于对归纳定义的关系所进行的讨论是比较典型的, 因此, 我们要详细地证明这种情况.

令 f 为项的哥德尔数集的特征函数，从“项”的定义，我们可得

$$f(a)=\begin{cases}1 & \text{如果 } a \text{ 是变元的哥德尔数,}\\ 1 & \text{如果 } (\exists i<\square, \exists k<a)[i \text{ 是一个数字序列，且}\\ & (\forall j<\mathrm{lh}i)f((i)_j)=1, \text{ 且 } k \text{ 是 } h \text{ 在某个 } (\mathrm{lh}i)\\ & \text{元函数符号上的取值并且 } a=\langle k\rangle * *_{j<\mathrm{lh}i}(i)_j],\\ 0 & \text{其他情况.}\end{cases}$$

但可以代替符号“□”的 i 的上界是多少呢？在讨论 f 的可表示性之前，我们需要 i 的一个上界，这取决于 a 的某种表示方式. 226

我们可以取 $i<a^{a\ \mathrm{lh}a}$. 为了证明这一点，假设 $a=\sharp st_1\cdots t_n$(其中 s 是一个 n 元函数符号，且 $t_1,\cdots,t_n$ 是项). 然后我们取 $i=\langle\sharp t_1,\cdots,\sharp t_n\rangle$. 根据 a，这个数有多大呢？我们可以得到它的上界为：

$$\begin{aligned} i&=2^{\sharp t_1+1}\cdots p_{n-1}^{\sharp t_n+1}\\ &\leqslant 2^a\cdots p_{n-1}^a\\ &<2^a\cdots p_{\mathrm{lh}a-1}^a \quad \text{因为 } n=\mathrm{lh}i<\mathrm{lh}a\\ &\leqslant a^a\cdots a^a \quad (\mathrm{lh}a\ \text{次})\ \text{因为 } a=2^{(a)_0+1}\cdots p_{\mathrm{lh}a-1}^{(a)_{\mathrm{lh}a-1}+1}\geqslant p_{\mathrm{lh}a-1}\\ &=(a^a)^{\mathrm{lh}a}=a^{a\mathrm{lh}a}\end{aligned}$$

因此在上面 f 的式子中，我们用 $a^{a\mathrm{lh}a}$ 代替 □.

尽管上述等式的右边用到了 f，但它只用到当 $(i)_j<a$ 时，$f((i)_j)$ 的值. 同时这个特点又使我们可以使用原始递归式. $f(a)=g(\bar{f}(a),a)$，其中

$$g(s,a)=\begin{cases}1 & \text{如果 } a \text{ 是变元的哥德尔数,}\\ 1 & \text{如果 } (\exists i<a^{a\mathrm{lh}a}, \exists k<a)[i\text{是一个数字序列，且}\\ & (\forall j<\mathrm{lh}i)(s)_{(i)_j}=1, \text{ 且 } k \text{ 是 } h \text{ 在某个 } (\mathrm{lh}i)\\ & \text{元函数符号上的取值且 } a=\langle k\rangle * *_{j<\mathrm{lh}i}(i)_j],\\ 0 & \text{其他情况.}\end{cases}$$

在这个式子中，如果让 s 等于 $\bar{f}(a)$，那么对于所有的 $(i)_j<a$，有 $(s)_{(i)_j}=f((i)_j)$. 这样由定理 33P 知，如果 g 是可表示的，那么 f 也是可表示的.

现在还需要证明 g 是可表示的，这一点可以从上一节的结论中直接得到. 简要地说，g 的图像是 3 个关系的并，这 3 个关系分别对应上面式子中的 3 个条件. 这三者都可以通过对等于以及其他的可表示关系通过添加有界量词和对可表示函数进行替换得到. ■

(3) 原子公式的哥德尔数组成的集合是可表示的.

证明 a 是一个原子公式的哥德尔数当且仅当$(\exists i<a^{a\mathrm{lh}a}, \exists k<a)[i$ 是数字序列且$(\forall j<\mathrm{lh}i)$ $(i)_j$ 是项的哥德尔数并且 k 是 h 在某个 $(\mathrm{lh}i)$ 元谓词符号上的取值，$a=\langle k\rangle * *_{j<\mathrm{lh}i}(i)_j]$. ■

(4) 合式公式的哥德尔数组成的集合是可表示的. 227

证明 我们已经知道合式公式是归纳定义的. 令 f 是该集合的特征函数，那么

$$f(a)=\begin{cases}1 & \text{如果 } a \text{ 是变元的哥德尔数,}\\ 1 & \text{如果}(\exists i<a)[a=\langle h((),h(\neg)\rangle*i*\langle h())\rangle\text{且}\\ & f(i)=1]\\ 1 & \text{如果}(\exists i,j<a)[a=\langle h(()\rangle*i*\langle h(\to)\rangle*\\ & j*\langle h())\rangle\text{且}f(i)=f(j)=1]\\ 1 & \text{如果}(\exists i,j<a)[a=\langle h(\forall)\rangle*i*j\text{且}\\ & i\text{是变元的哥德尔数并且}f(j)=1]\\ 0 & \text{其他情况.}\end{cases}$$ ■

我们可以采用对项的哥德尔数集相同的讨论来得到 f 的可表示性.

(5) 存在可表示函数 Sb 使得对于一个项或公式 α, 变元 x, 及项 t,

$$\mathrm{Sb}(\sharp\alpha,\sharp x,\sharp t)=\sharp\alpha^x_t.$$

证明　我们需要利用 $\mathrm{Sb}(i,b,c)$, $i<a$ 的值来定义 $\mathrm{Sb}(a,b,c)$. 和编目 2 中 (项的集合的特征函数) 的情况一样, 我们来证明 $\overline{\mathrm{Sb}}$ 和 Sb 都是可表示的.

函数 Sb 可以由下面 6 个子句来描述.

(i) 如果 a 是一个变元的哥德尔数并且 $a=b$, 那么

$$\mathrm{Sb}(a,b,c)=c.$$

(ii) 如果 $(\exists i<a^{\mathrm{alh}a},\exists k<a)(i$ 是一个数字序列且 $(\forall j<\mathrm{lh}i)(i)_j$ 是项的哥德尔数且 k 是 h 在某个 $(\mathrm{lh}i)$ 元函数符号或谓词符号上的值, $a=\langle k\rangle**_{j<\mathrm{lh}i}(i)_j)$, 则对于 i 和 k,

$$\mathrm{Sb}(a,b,c)=\langle k\rangle**_{j<\mathrm{lh}i}\mathrm{Sb}((i)_j,b,c)$$

(iii) 如果 $(\exists i<a)[i$ 是一个合式公式的哥德尔数且 $a=\langle h((),h(\neg)\rangle*i*\langle h())\rangle]$, 则对于 i,

$$\mathrm{Sb}(a,b,c)=\langle h((),h(\neg)\rangle*\mathrm{Sb}(i,b,c)*\langle h())\rangle \tag{3-1}$$

(iv) 如果 $(\exists i,j<a)[i,j$ 是合式公式的哥德尔数且 $a=\langle h(()*i*h(\to)\rangle*j*\langle h())\rangle]$, 则对 i 和 j,

228

$$\mathrm{Sb}(a,b,c)=\langle h(()\rangle*\mathrm{Sb}(i,b,c)*\langle h(\to)\rangle*\mathrm{Sb}(j,b,c)*\langle h())\rangle$$

(v) 如果 $(\exists i,j<a)[i$ 是变元的哥德尔数且 $i\neq b$ 且 j 是合式公式的哥德尔数且 $a=\langle h(\forall)\rangle*i*j]$, 则对于 i,j,

$$\mathrm{Sb}(a,b,c)=\langle h(\forall)\rangle*i*\mathrm{Sb}(j,b,c)$$

(vi) 如果 a,b 不符合上述所列的条件 (此时我们不考虑 $\mathrm{Sb}(a,b,c)$ 的显式表达式), 则

$$\mathrm{Sb}(a,b,c)=a.$$

至此, 函数 Sb 可以由原始递归式得到

$$\mathrm{Sb}(a,b,c)=G(\overline{\mathrm{Sb}}(a,b,c),a,b,c)$$

其中 G 是 4 元函数. G 的图像是 6 个 5 元关系的并

$$G = R_1 \cup R_2 \cup R_3 \cup R_4 \cup R_5 \cup R_6$$

6 个关系分别对应于上面的 6 个条件.

第一个关系为

$$R_1 = \{\langle s, a, b, c, d\rangle | a\text{是变元的哥德尔数且}a = b, d = c\}$$

第二个关系为

$$R_2 = \{\langle s, a, b, c, d\rangle | (\exists i < a^{a\text{lh}a}, \exists k < a)[i\text{是一个数字序列且}(\forall j < \text{lh}i)(i)_j \text{ 是项的哥德尔数且 } k \text{ 是 } h \text{ 在某个 (lh}i\text{) 元函数符号或谓词符号上的值且 } a = \langle k\rangle * *_{j<\text{lh}i}(i)_j, d = \langle k\rangle * *_{j<\text{lh}i}(s)_{(i)_j}]\}$$

并且其他的关系可以类似地将描述 Sb 的语句对应地翻译过来.

我们必须注意到, G 确实是一个函数, 它是单值的. 这是因为一个数 a 不可能同时满足两个条件. 例如, 如果 a 满足条件 (ii), 则从 2.3 节可以知道 i 和 k 的值是唯一确定的.

最后, 我们可以用常用的办法证明 $R_1 \sim R_6$ 是可表示的, 所以 G 是可表示的, 进而 $\overline{\text{Sb}}$ 和 Sb 都是可表示的. (替换是一种复杂的运算) ■

(6) 在 n 上取值为 $\sharp(\mathbf{S}^n\mathbf{0})$ 的函数是可表示的.

证明 我们把这个函数称为 f, 则

$$f(0) = \langle h(\mathbf{0})\rangle,$$
$$f(n+1) = \langle h(\mathbf{S})\rangle * f(n).$$

再利用上节中的习题 8 就可以证明了. ■

(7) 存在一个可表示关系 Fr 使得对于一个项或公式 α, 及变元 x, 229

$$\langle \sharp\alpha, \sharp x\rangle \in \text{Fr} \Leftrightarrow x \text{ 在 } \alpha \text{ 中自由出现}$$

证明 $\langle a, b\rangle \in \text{Fr} \Leftrightarrow \text{Sb}(a, b, \sharp\mathbf{0}) \neq a$ ■

(8) 全体句子的哥德尔数组成的集合是可表示的.

证明 a 是一个句子的哥德尔数当且仅当 a 是一个公式的哥德尔数并且对于任意的 $b < a$, 如果 b 是一个变元的哥德尔数, 则 $\langle a, b\rangle \notin \text{Fr}$. ■

(9) 存在一个可表示的关系 Sb1, 使得对于公式 α, 变元 x 和项 t, $\langle \sharp\alpha, \sharp x, \sharp t\rangle \in \text{Sb1}$ 当且仅当 α 中的 x 可由 t 替换.

证明 习题 1. ■

(10) 关系 Gen 是可表示的, $\langle a, b\rangle \in$Gen 当且仅当 a 是一个公式的哥德尔数并且 b 是对这个公式运用推广法则后得到的公式 (以后简称推广式) 的哥德尔数.

证明 $\langle a, b\rangle \in$Gen 当且仅当 $a = b$ 或 $(\exists i, j < b)[i$ 是变元的哥德尔数并且 $\langle a, j\rangle \in$Gen 同时 $b = \langle\langle h(\forall)\rangle * i * j]$. 再对 Gen 的特征函数进行常规的讨论就行了. ■

(11) 重言式的哥德尔数组成的集合是可表示的.

我们可以用真值表的方法不规范地判定一个集合是否是重言式的集合. 为了得到可表示性, 我们根据哥德尔数对真值表进行了修改. 下面是几个简要的步骤:

(11.1) 关系 R, $\langle a,b\rangle \in R$ 当且仅当 a 是公式 α 的哥德尔数, 且 b 是 α 的基本组成的哥德尔数, 则关系 R 是可表示的.

证明　$\langle a,b\rangle \in R \Leftrightarrow a$ 是一个公式的哥德尔数并且下列之一成立:

(i) $a=b$ 且 $(a)_0 \neq h(()$.

(ii) $(\exists i<a)[a=\langle h((),h(\neg)\rangle * i * \langle h())\rangle$ 且 $\langle i,b\rangle \in R]$.

(iii) 与 (ii) 类似, 只需将 $\neg$ 改为 $\to$.

再对 R 的特征函数进行常规的讨论就行了. ■

(11.2) 存在可表示的函数 P 使得对于一个公式 α, $P(\sharp\alpha)=\langle \sharp\beta_1,\cdots,\sharp\beta_n\rangle$, 即将 α 所有的基本组成部分的哥德尔数按照顺序排列出来.

证明　首先, 我们定义函数 g 用以确定 $\natural a$ 中 $\natural y$ 之后的基本组成 (这里 $\natural a$ 表示哥德尔数为 a 的公式 α, 即 $a=\sharp\alpha$).

$$g(a,y)=n\text{使得}n=a+1\text{或者 } y<n \text{ 且 } \langle a,n\rangle \in R \text{ 的最小元素}.$$

接下去定义函数 h, 使得 $h(a,n)$ 给出 $\natural a$ 的第 $(n+1)$ 个基本组成 (如果存在这么多个基本组成):

230

$$h(a,0)=g(a,0) \qquad h(a,n+1)=g(a,h(a,n)).$$

最后, 令 $P(a)=*_{i<k}\langle h(a,i)\rangle$, 其中 k 是使得 $h(a,k)>a$ 的最小元素. ■

(11.3) 我们称整数 v 是 α 的*真值指派编码*, 当且仅当 v 是一个数字序列, $\mathrm{lh}v=\mathrm{lh}P(\sharp\alpha)$ 并且 $(\forall i<\mathrm{lh}v)(\exists e<2)(v)_i=\langle (P(\sharp\alpha))_i,e\rangle$. 这是一个 $v,\sharp\alpha$ 上的可表示关系.

例如, 如果 $P(\sharp\alpha)=\langle \sharp\beta_0,\cdots,\sharp\beta_n\rangle$, 则

$$v=\langle\langle \sharp\beta_0,e_0\rangle,\cdots,\langle \sharp\beta_n,e_n\rangle\rangle,$$

其中每个 e_i 是 0 或 1. 我们要根据 $\sharp\alpha$ 的值来估计 v 的上界. 当每个 e_i 都取 1 时, v 有最大值, 同时由于 $\sharp\beta_i \leqslant \sharp\alpha$, 因此

$$\begin{aligned} v &\leqslant \langle\langle \sharp\alpha,1\rangle,\cdots,\langle \sharp\alpha,1\rangle\rangle \\ &= *_{i<\mathrm{lh}P(\sharp\alpha)}\langle\langle \sharp\alpha,1\rangle\rangle. \end{aligned}$$

(11.4) 存在一个可表示的关系 Tr 使得对于一个公式 α 和 α(或更多) 的真值指派编码 v, $\langle \sharp\alpha,\ v\rangle \in$Tr 当且仅当真值指派满足 α.

证明　习题 2. ■

最后, α 是个重言式当且仅当 α 是个公式且对于 α 的真值指派的每个编码 v, $\langle \sharp\alpha,\ v\rangle \in$ Tr. 正如 11.3 中解释的那样, (自然语言中)v 的量词可以由 $\sharp\alpha$ 的可表示函数来约束.

(12) 由具有形式为 $\forall x\varphi \to \varphi_t^x$ 的公式的哥德尔数组成的集合是可表示的, 其中 t 是可以替换变元 x 的项.

证明 公式 α 具有上述形式当且仅当 ($\exists$ 合式公式 $\varphi < \alpha$) ($\exists$ 变元 $x < \alpha$)($\exists$ 项 $t < \alpha$)[t 可替换 φ 中的 x 且 $\alpha = \forall x\varphi \to \varphi^x_t$]. 此处 "$\varphi < \alpha$" 是指 $\sharp\varphi < \sharp\alpha$. 用哥德尔数来描述这个等价条件也是很容易的：$a$ 属于该集合当且仅当 $(\exists f < a)(\exists x < a)(\exists t < a)$ [f 是一个公式的哥德尔数且 x 是变元的哥德尔数，t 是项的哥德尔数且 $\langle f, x, t\rangle \in \text{Sb1}$ 且 $a = \langle h((), h(\forall)\rangle * x * f * \langle h(\to)\rangle * \text{Sb}(f, x, t) * \langle h())\rangle$]. ■

(13) 由具有形式为 $\forall x(\alpha \to \beta) \to \forall x\alpha \to \forall x\beta$ 的公式的哥德尔数组成的集合是可表示的.

证明 公式 γ 具有上述形式当且仅当 ($\exists$ 变元 $x < \gamma$) ($\exists$ 公式 $\alpha, \beta < \gamma$)[$\gamma = \forall x(\alpha \to \beta) \to \forall x\alpha \to \forall x\beta$]. 和在 12 中一样，这个等价式很容易用哥德尔数来重述. ■

(14) 由具有形式为 $\alpha \to \forall x\alpha$ 的公式的哥德尔数组成的集合是可表示的，其中 x 在 α 中不自由出现.

证明 与 (13) 类似. ■

(15) 由具有形式为 $x = x$ 的公式的哥德尔数组成的集合是可表示的. 231

证明 与 (13) 类似. ■

(16) 由具有下列形式为 $x = y \to \alpha \to \alpha'$ 的公式的哥德尔数组成的集合是可表示的，其中 α 是原子公式，将 α 中零处或多处出现的 x 用 y 代替就得到 α'.

证明 除了"部分替换"关系外，与 (13) 类似. 令 $\langle a, b, x, y\rangle \in$Psb 当且仅当 x 和 y 是变元的哥德尔数，a 是原子公式的哥德尔数，b 是长度为 lh a 的数字序列，且对于所有的 $j < \text{lh}\ a$, $(a)_j = (b)_j$ 或 $(a)_j = x$ 且 $(b)_j = y$. 这个关系是可表示的. ■

(17) 逻辑公理的哥德尔数集是可表示的.

证明 α 是逻辑公理当且仅当 $\exists\beta \leqslant \alpha$ 使得 α 是 β 的推广式且 β 是在 11~16 条中的集合中的公式. ■

(18) 对于公式的有限集合 A,

$$\{\mathcal{G}(D) | D \text{ 是 } A \text{ 的推论}\}$$

是可表示的. 实际上只需要推出 $\sharp A$ 是可表示的就足够了.

证明 数 d 属于这个集合当且仅当 d 是长度为正的数字序列且对于每个小于 lh$d$ 的数 i，下面三个条件之一成立.

(1) $(d)_i \in \sharp A$,

(2) $(d)_i$ 是逻辑公理的哥德尔数，或者

(3) $(\exists j, k < i)[(d)_j = \langle h(()\rangle * (d)_k * \langle h(\to)\rangle * (d)_i * \langle h())\rangle]$.

只要 $\sharp A$ 是可表示的，则上述条件就是可表示的. 对于有限集合 A，情况也是一样的. ■

(19) 任意递归关系在 Cn A_E 中是可表示的.

证明 我们知道关系 R 是递归的当且仅当存在某个和谐的有限句子集使得某个公式 ρ 能在 Cn A_E 中表示 R . (不失一般性，我们假设语言中只有有限多个参数：它们包含在有限集 A, ρ 以及 $\mathbf{0}, \mathbf{S}$ 和 $\forall$ 中.) 当 R 是一元关系时，我们有 $a \in R$ 当且仅当 D 是由 $\rho(\mathbf{S}^a\mathbf{0})$ 或 $\neg\ \rho(\mathbf{S}^a\mathbf{0})$ 的 A 递归得到的最小推论.

更形式化地说，$a \in R$ 当且仅当 $f(a)$ 的最后一个元素是 $\sharp\rho(\mathbf{S}^a\mathbf{0})$，其中

$$f(a) = \text{使得 } d \text{ 在 18 条的集合中并且它的最后一个元素是 } \sharp\rho(\mathbf{S}^a\mathbf{0}) \text{ 或 } \sharp\neg\ \rho(\mathbf{S}^a\mathbf{0}).$$

对于这个 (固定的)ρ，总有这样的 d 存在. ■

由于 19 条的逆命题是显然的，因此我们有

[232] **定理 34A**　一个关系是递归的当且仅当它在理论 $\mathrm{Cn}\ A_E$ 中是可表示的.

因此我们通常不使用“可表示”，而是使用“递归”这个词.

推论 34B　任意递归关系在 $\mathfrak{N}$ 中是可定义的.

(20) 现在假设我们有句子集 A 使得 $\sharp A$ 是递归的. 则 $\sharp\mathrm{Cn}\ A$ 不是递归的 (我们在下一节中将会证明)，但是我们仍然有一个方法由 A 来定义 $\mathrm{Cn}\ A$:

$$a \in \sharp\mathrm{Cn}\ A \text{ iff } \quad \exists d[d\text{是由 } A \text{ 递归的一个数并且 } d \text{ 的最后一个组成是 } a \text{ 同时 } a \text{ 是一个句子的哥德尔数}]$$

由 18 的证明可知，方括号内的内容是递归的. 但一般情况下，我们不可能求出 d 的上下界. 我们只能说 $\sharp\mathrm{Cn}A$ 是递归关系的论域 (或者说，是*可递归枚举的*. 这个概念将在后面学到).

第 20 条将在后面的内容中发挥关键的作用，而且，它还会在定理 35I 中得到重申.

(21) 如果 $\sharp A$ 是递归的并且 $\mathrm{Cn}A$ 是完全理论，那么 $\sharp\mathrm{Cn}A$ 是递归的.

换句话说，一个可递归公理化的完全理论是递归的. 这和推论 25G 的结果很类似，推论 25G 断言一个可公理化的完全理论是可判定的.

它的证明在本质上没有改变. 令 (在和谐的情况下)

$$g(s) = \text{最小的 } d\text{，使得 } s \text{ 不是句子的哥德尔数，或 } d \text{ 在 18 条的集合中并且 } d \text{ 的最后一个组成是 } s \text{ 或} \langle h((), h(\neg)\rangle * s * \langle h())\rangle.$$

这样 $g(\sharp\sigma)$ 是从 A 得到的 σ 或 $(\neg\ \sigma)$ 的最小递归 $\mathcal{G}$. 并且 $s \in \sharp\mathrm{Cn}\ A$ 当且仅当 $s > 0$ 并且 $g(s)$ 的最后组成部分是 s.

从这一点上看，我们应该重新考虑丘奇论题的合理性. 假设关系 R 是可判定的，那么判定过程一定存在一个有限的程序指令. 这个过程本身大概由几个基本步骤组成，然后被反复执行. (熟悉计算机的读者都知道一个很短的程序可能需要较长的运行时间，一些命令会被反复地使用.) 每个基本步骤可能都很简单.

[233] 通过使用哥德尔数，我们可以在整数中来反映判定过程. R 的特征函数具有下面形式:

$K_R(\vec{a}) = U$ [最小的 s 使得

(i) $(s)_0$ 解码输入元素 $\vec{a}$;

(ii) 对于所有正数 $i < \mathrm{lh}s$，$(s)_i$ 由 $(s)_{i-1}$ 通过执行基本步骤得到;

(iii) s 的最后一个组成部分描述了计算结束的最后状态],

其中 U(结果函数) 是某个简单函数，它从 s 的最后一个组成部分里得出 (肯定的或否定的) 答案. R 的递归性归结为 U 的递归性和 (i), (ii), (iii) 中关系的递归性. 在特殊情况下，

例如判定过程是由 3.6 节中的寄存器产生的，这些组成部分的递归性很容易检验. 我们很难想象，如果一个判定过程中含有不可递归的成分，但最终它却是有效的判定. 例如在 (ii) 中，我们必须能使每个基本步骤都非常简单，特别要保证它们是递归的.

习题

1. 证明本节中的第 9 条.
2. 证明本节中的第 11.4 条.
3. 利用 3.3 节中习题 11 的结果来证明项的哥德尔数集是可表示的 (第 2 条).
4. 设 T(在含有 $\mathbf{0}$ 和 $\mathbf{S}$ 的有限递归语言中) 是一个可递归公理化的和谐理论，证明 T 中任意的可表示关系必然是递归的.

3.5 不完全性和不可判定性

3.3 节和 3.4 节中的内容将在这一节中发挥重要的作用. 我们已经把哥德尔数指派给了表达式，并且还证明了 $\mathbb{N}$ 上的某些可递归判定的关系 (与表达式的语法概念有关) 在 Cn A_E 中是可表示的.

在这一整节中，我们都假设在问题中的语言都是 $\mathfrak{N}$ 的语言. (这影响到 "Cn" 和 "理论" 的意思.)

不动点引理 对于只含有自由变元 v_1 的公式 β，我们可以找到句子 σ 使得

$$A_E \vdash [\sigma \leftrightarrow \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})].$$

我们可以认为 σ 间接地表达 "β 是真的我. " 当然，实际上 σ 什么也没说，它只是一些符号串而已. 甚至当我们在 $\mathfrak{N}$ 中把它翻译成自然语言时，它也仅仅是关于一些数字和它们的后继及运算结果的句子. 正是因为我们把数字和表达式联系起来，我们才能把 σ 看作一个公式. 234

证明 设 $\theta(v_1, v_2, v_3)$ 在 Cn A_E 中函数表示一个函数，这个函数在 $\langle\sharp\alpha, n\rangle$ 的值为 $\sharp(\alpha(\mathbf{S}^n\mathbf{0}))$. (见 3.4 节中 5 和 6 两条.) 我们首先考虑公式

$$\forall v_3[\theta(v_1, v_2, v_3) \to \beta(v_3)]. \tag{1}$$

(假设 β 中的 v_1 由 v_3 代入，则上述式子中只有 v_1 是自由变元. 它定义了 $\mathfrak{N}$ 中的一个集合，$\sharp\alpha$ 属于这个集合当且仅当 $\sharp(\alpha(\mathbf{S}^{\sharp\alpha}\mathbf{0}))$ 在 β 定义的集合中.) 设 q 是 (1) 的哥德尔数，σ 为

$$\forall v_3[\theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3) \to \beta(v_3)].$$

σ 是将式 (1) 中的 v_1 用 $\mathbf{S}^q\mathbf{0}$ 代替之后得到的. 我们必须注意 σ 并没有断定 (在 $\mathfrak{N}$ 下)$\sharp\sigma$ 在 β 定义的集合中. 我们必须验证

$$\sigma \leftrightarrow \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}) \tag{2}$$

是 A_E 的推论. 由于被 θ 函数表示的函数在 $\langle q, q\rangle$ 的值是 $\sharp\sigma$，因此，有

$$A_E \vdash \forall v_3[\theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3) \leftrightarrow v_3 = \mathbf{S}^{\sharp\sigma}\mathbf{0}]. \tag{3}$$

我们可以依照下列方法得到 (2) 式：

(→) (通过观察 σ) 我们容易证明

$$\sigma \vdash \theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^{\sharp\sigma}) \to \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}).$$

同时，根据式 (3)，有

$$A_E \vdash \theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, \mathbf{S}^{\sharp\sigma}\mathbf{0}).$$

因此

$$A_E; \sigma \vdash \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}),$$

这证明了式 (2) 的一半.

← 式 (3) 中的句子蕴涵

$$\beta(\mathbf{S}^{\sharp\sigma}\mathbf{0}) \to [\forall v_3(\theta(\mathbf{S}^q\mathbf{0}, \mathbf{S}^q\mathbf{0}, v_3) \to \beta(v_3))].$$

而方括号内的公式恰好正是 σ. ■

(有时，我们用 $\ulcorner\sigma\urcorner$ 表示 $\mathbf{S}^{\sharp\sigma}\mathbf{0}$. 如果使用这个符号，那么不动点引理可以简写为 $A_E \vdash (\sigma \leftrightarrow \beta(\ulcorner\sigma\urcorner))$.)

这个引理的第一个应用与 Cn A_E 的子理论没有关系，而只需要下面这个相对较弱的结果

235

$$\models_{\mathfrak{N}} [\sigma \leftrightarrow \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})].$$

塔斯基不可定义定理 (1933) 集合 $\sharp$ Th $\mathfrak{N}$ 在 $\mathfrak{N}$ 中是不可定义的.

证明 我们考虑任意的公式 β(假定它能定义 $\sharp$ Th $\mathfrak{N}$)，根据不动点引理 (将其运用到 $\neg\beta$ 上) 我们有句子 σ 使得

$$\models_{\mathfrak{N}} [\sigma \leftrightarrow \neg\beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})].$$

(如果 β 确实定义了 $\sharp$ Th $\mathfrak{N}$，则 σ 就间接地表示“我是错的.”) 则

$$\models_{\mathfrak{N}} \sigma \Leftrightarrow \not\models_{\mathfrak{N}} \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})],$$

因此，要么 σ 是真的，但 (它的哥德尔数) 不在 β 定义的集合中，要么 σ 是假的且在那个集合中. 无论哪种情况，σ 都表示 β 不能定义 $\sharp$ Th $\mathfrak{N}$. ■

上述定理立刻就说明了 $\mathfrak{N}$ 的理论是不可判定的.

推论 35A $\sharp$ Th $\mathfrak{N}$ 不是递归的.

证明 (根据推论 34B) 任意的递归集在 $\mathfrak{N}$ 中都是可定义的. ■

哥德尔不完全性定理 (1931) 如果 $A \subseteq$ Th $\mathfrak{N}$，并且 $\sharp A$ 是递归的，则 Cn A 不是一个完全理论.

这样，Th $\mathfrak{N}$ 就不可能是完全可递归公理化的.

证明 由于 $A \subseteq$ Th $\mathfrak{N}$，因此我们有 Cn $A \subseteq$ Th $\mathfrak{N}$. 如果 Cn A 是完全理论，则等号成立. 但是，如果 Cn A 是完全理论，$\sharp$Cn A 就是递归的 (根据上一节的 21 条). 我们由上面的推论可知，$\sharp$ Th $\mathfrak{N}$ 不是递归的. ■

我们特别要指出，Cn A_E 不是完全理论，因此不等于 Th $\mathfrak{N}$. 并且我们不可能通过增加任何公理的递归集来消除不完全性. (所谓句子的递归集是指集合 Σ 使得 $\sharp\Sigma$ 是递归的.)

我们可以从哥德尔定理的证明中得到更多的东西. 想像一个特别的递归集 $A \subseteq \text{Th}\,\mathfrak{N}$, 根据 3.4 中的 20 条，我们可以找到一个公式 β, 它在 $\mathfrak{N}$ 中定义了 $\sharp\text{Cn}\,A$. 按照塔斯基定理的证明过程构造的句子 σ 是一个真的句子但不在 Cn A 中. 这个句子断言 $\sharp\sigma$ 不属于 β 定义的集合，即这个句子间接地表示，"我不是 A 的定理. " 这样 $A \nvdash \sigma$, 当然，$A \nvdash \neg\,\sigma$. 这个证明方法和哥德尔的原始证明很相近，没有使用塔斯基定理. 正是因为这个原因，哥德尔在叙述他的定理时，没有将 Th $\mathfrak{N}$ 包括进去；我们冒昧地对定理进行了改动.

236

在下面的定理证明中，我们需要一个引理. 这个引理 (概括地) 说，我们可以向递归理论中添加一个新的公理 (和有限多个新的公理)，却不改变理论的递归性.

引理 35B 如果 $\sharp\text{Cn}\,\Sigma$ 是递归的，则 $\sharp\text{Cn}(\Sigma;\tau)$ 是递归的.

证明 $\alpha \in \text{Cn}(\Sigma;\tau) \Leftrightarrow (\tau \to \alpha) \in \text{Cn}\,\Sigma$. 这样就有

$\alpha \in \sharp\text{Cn}(\Sigma;\tau) \Leftrightarrow \alpha$ 是一个句子的哥德尔数，并且 $\langle h(()\rangle * \sharp\tau * \langle h(\to)\rangle * \alpha * \langle h())\rangle$ 在 $\sharp\text{Cn}\,\Sigma$ 中.

这可以由上节课的结果递归得到. ■

定理 35C(Cn $\mathbf{A_E}$的强不可判定性) 设 T 是一个理论使得 $T \cup A_E$ 是和谐的，则 $\sharp T$ 不是可递归的.

(请注意，本节中提到的语言都是 $\mathfrak{N}$ 的语言，因此这个定理中的"理论"指的是"$\mathfrak{N}$ 的语言的理论".)

证明 设 T' 为 $\text{Cn}(T \cup A_E)$ 的理论. 如果 $\sharp T$ 是递归的，又因为 A_E 是有限的，因此由上面一个引理我们能够得到，$\sharp T'$ 也是递归的.

如果 $\sharp T'$ 是递归的，那么它在 Cn A_E 中可以由某个公式 β 表示. 由不动点引理可知，可以找到一个句子 σ, 使得

$$A_E \vdash [\sigma \leftrightarrow \neg\,\beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})]. \tag{$*$}$$

(σ 间接表示，"我不在 T' 中. ")

$$\begin{aligned}\sigma \notin T' &\Rightarrow \sharp\sigma \notin \sharp T'\\ &\Rightarrow A_E \vdash \neg\,\beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})\\ &\Rightarrow A_E \vdash \sigma \qquad \text{由 } (*)\\ &\Rightarrow \sigma \in T'.\end{aligned}$$

因此，得到 $\sigma \in T'$. 但这个结论也是站不住脚的：

$$\begin{aligned}\sigma \in T' &\Rightarrow \sharp\sigma \in \sharp T'\\ &\Rightarrow A_E \vdash \beta(\mathbf{S}^{\sharp\sigma}\mathbf{0})\\ &\Rightarrow A_E \vdash \neg\,\sigma \qquad \text{由 } (*)\\ &\Rightarrow (\neg\,\sigma) \in T',\end{aligned}$$

这与 T' 的和谐性相矛盾. ■

推论 35D 假设 $\sharp\Sigma$ 是递归的, $\Sigma \cup A_E$ 是和谐的, 则 $\text{Cn}\ \Sigma$ 不是完全理论.

证明 已知可递归公理化的完全理论是递归的 (3.4 节中 21 条). 但根据上面的定理, $\sharp\text{Cn}\ \Sigma$ 不是递归的. ■

这个推论也是哥德尔不完全性定理的一种表述, 只是用 A_E 的和谐性代替了"在 $\mathfrak{N}$ 中为真"这一条件.

237

丘奇定理 (1936) (在 $\mathfrak{N}$ 的语言中) 取值为真的句子的哥德尔数的集合不是递归的.

证明 在 $\text{Cn}\ A_E$ 的强不可判定性的定理中, 取 T 为语言的最小理论 (取值为真的句子集) 即可. ■

取值为真的合式公式的哥德尔数集合也是不可递归的, 否则, 取值为真的句子的哥德尔数的集合就是递归的.

这个证明可以运用到 $\mathfrak{N}$ 的语言上. 对于一个含有更多参数的语言来说, 取值为真的句子的集合仍然不是可递归的 (否则它与 $\mathfrak{N}$ 的语言的交集是递归的). 实际上, 语言中只要含有至少一个二元谓词符号就够了. (见推论 37G.) 另一方面, 语言的下界必须要确定. 如果语言 (等号的语言) 中只含有 $\forall$ 一个参数, 则取值为真的公式的集合就是可判定的. (见习题 6.) 更一般地, 如果参数只有 $\forall$ 和一元谓词符号, 则取值为真的公式的集合是可判定的.

3.5.1 递归可枚举性

自然数上的一个关系是*递归可枚举的*当且仅当它具有形式

$$\{\vec{a} \mid \exists b \langle \vec{a}, b\rangle \in Q\}$$

其中 Q 是递归的. 递归可枚举关系在逻辑中占有重要的地位. 它们是能行可枚举关系的形式表示 (不久将要给以解释).

(递归可枚举的标准缩写为 "r.e." . 如果在这个定义中用"可计算"代替"递归", 则称为*可计算枚举关系*, 可计算枚举缩写为 c.e..)

和递归关系一样, 递归可枚举关系可以在 $\mathfrak{N}$ 中定义. 如果 $\varphi(v_1, v_2)$ 在 $\mathfrak{N}$ 中定义了一个二元关系 Q, 则 $\exists v_2 \varphi(v_1, v_2)$ 定义了 $\{a \mid \exists b \langle a, b\rangle \in Q\}$.

定理 35E 下述关于 m 元关系 R 的结论是等价的:

(1) R 是递归可枚举的.

(2) R 是某个递归函数 Q 的定义域.

(3) 对于某个递归的 $(m+1)$ 元关系 Q,

$$R = \{\langle a_1, \cdots, a_m\rangle \mid \exists b \langle a_1, \cdots, a_m, b\rangle \in Q\}.$$

(4) 对于某个递归的 $(m+n)$ 元关系 Q,

$$R = \{\langle a_1, \cdots, a_m\rangle \mid \exists b_1, \cdots, b_n \langle a_1, \cdots, a_m, b_1, \cdots, b_n\rangle \in Q\}.$$

238

证明 根据定义, 1 和 3 是等价的. 同时根据定义域和 $(m+1)$ 元组的概念 (第 0 章), 2

和 3 也是等价的. 3 推出 4 是显然的. 因此, 我们只要证明 4 能推出 3 就可以了. 由于

$$\exists b_1, \cdots, b_n \langle a_1, \cdots, a_m, b_1, \cdots, b_n \rangle \in Q$$
$$\text{iff} \quad \exists c \langle a_1, \cdots, a_m, (c)_0, \cdots, (c)_{n-1} \rangle \in Q$$

并且 $\{\langle a_1, \cdots, a_m, c \rangle | \langle a_1, \cdots, a_m, (c)_0, \cdots, (c)_{n-1} \rangle \in Q\}$ 是递归的, 其中 Q 是递归的. (这里我们使用序列解码函数把量词串转化成一个数字.) ■

根据这个定理的第 4 个等价条件我们可以知道, R 是递归可枚举的当且仅当它可以由 $\mathfrak{N}$ 中的公式 $\exists x_1, \cdots, \exists x_n \varphi$ 定义, 这里 φ 由 A_E 数字确定. 实际上, 这里我们可以要求 φ 是无量词的, 这个结果于 1961 年 (含有幂乘运算) 和 1970 年 (不含幂乘运算) 被证明. 证明的过程包含一些数论的知识, 因此我们忽略它们的证明.

请注意, 任意递归关系也是递归可枚举的, 因为如果 R 是递归的, 则它被 $\mathfrak{N}$ 中的公式 $\exists x_1 \cdots \exists x_n \varphi$ 定义, 其中 φ 由 A_E 数字确定, 且 $x_1, \cdots, x_n$ 不在 φ 中出现.

定理 35F 一个关系是递归的当且仅当它和它的补集都是递归可枚举的.

我们已经知道一个关系是可判定的当且仅当它和它的补集都是能行可枚举的. 上述的定理正是这个事实 (定理 17F) 形式表述的.

证明 如果一个函数是递归的, 则它的补也是递归的, 进而, 它们都是递归可枚举的.

反之, 假设 P 和它的补集都是递归可枚举的, 那么, 对于任意的 $\vec{a}$, 存在递归的 Q 和 R, 使得

$$\vec{a} \in P \Leftrightarrow \exists b \langle \vec{a}, b \rangle \in Q$$
$$\vec{a} \notin P \Leftrightarrow \exists b \langle \vec{a}, b \rangle \in R$$

令 $f(\vec{a})$ = 最小的 b, 使得 $\langle \vec{a}, b \rangle \in Q$ 或 $\langle \vec{a}, b \rangle \in R$. 这样的 b 总是存在的, 并且 f 是递归的. 最后,

$$\vec{a} \in P \Leftrightarrow \langle \vec{a}, f(\vec{a}) \rangle \in Q,$$

因此, P 是递归的. ■

递归可枚举关系是能行可枚举关系的形式化表示. 因此, 我们有下面非正式的结果, 它与定理 35E 中给出的递归可枚举性的特征是相互平行的. 239

***引理 35G** 一个关系是能行可枚举的当且仅当它是一个可判定关系的定义域.

证明 假设 Q 是由某个程序能行枚举的. 则 $\vec{a} \in Q$ 当且仅当 $\exists n[\vec{a}$ 在第 n 步枚举中出现]. 方括号中定义的关系是可判定的并且定义域为 Q.

反过来, 对于可判定的 R, 我们列举出集合 $\{\langle a, b \rangle | \exists n \langle a, b, n \rangle \in R\}$, 并且对于 $m = 0, 1, 2, \cdots$, 我们逐个检查是否 $\langle (m)_0, (m)_1, (m)_2 \rangle \in R$. 只要答案是肯定的, 我们就把 $\langle (m)_0, (m)_1 \rangle$ 作为输出结果. ■

推理 35H(丘奇论题, 第二形式) 一个关系是能行可枚举的当且仅当它是递归可枚举的.

证明　通过把可判定关系和递归关系等同起来，我们自然可以把可判定关系的定义域和递归关系的定义域看作是相同的. ■

实际上，丘奇论题的第二形式和第一形式是等价的. 为了从第二形式证明第一形式，我们要用到定理 35F 和 17F.

我们已经证明了可递归公理化的理论是递归可枚举的，但我们用的是另一种说法. 在这里我们重申一下这个结果，因为它表明了递归可枚举性在逻辑中所起的作用.

定理 35I　如果 A 是句子集，使得 $\sharp A$ 是递归的，则 $\sharp\text{Cn } A$ 是递归可枚举的.

证明　见 3.4 节中 20 条. ■

特别地，$\sharp\text{Cn } A_E$ 是递归可枚举的，但 (根据定理 35C) 它不是递归的. 在下一节中，我们还将看到是递归可枚举的但不是递归的集合的例子.

我们已经知道，一个理论如果有可判定的公理集，那么它是能行可枚举的 (推论 25F 和 26I)，上面的定理是这个不正式的结果的准确表述. 它表明，一个公理化的理论中可证明的结果和结构中取值为真的结果之间是有差距的. 用公理的递归集合，我们只可能得到推论的递归可枚举集. 然而根据塔斯基定理，Th $\mathfrak{N}$ 甚至不能在 $\mathfrak{N}$ 中定义，这要比递归可枚举弱得多.

既使我们扩展语言或者添加新的公理，这样的现象仍然存在. 只有当我们能够递归地区分演绎和非演绎的时候，定理的集合才可以被递归枚举出来. 例如，数论的句子集在我们通常使用的公理集合论系统中是可证明的，同时也是递归可枚举的. 而且，这个集合包含 A_E 并且是和谐的 (除非你所使用的系统非常奇怪). 这样，集合论不是递归的也不是完全的. (我们将在 3.7 节中更详细地讨论这个问题.)

240

3.5.2　弱可表示性

我们考虑递归可枚举集 Q，其中对于递归 R

$$a \in Q \Leftrightarrow \exists b \langle a, b\rangle \in R$$

我们知道存在公式 ρ 在 $\text{Cn } A_E$ 中表示 R. 因此，公式 $\exists v_2 \rho$ 在 $\mathfrak{N}$ 中定义了 Q. 这个公式在 $\text{Cn } A_E$ 中不能表示 Q，除非 Q 是递归的. 但我们说这个公式几乎可以表示 Q.

$$\begin{aligned} a \in Q &\Rightarrow \langle a, b\rangle \in R && \text{对于某个 } b \\ &\Rightarrow A_E \vdash \rho(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}) && \text{对于某个 } b \\ &\Rightarrow A_E \vdash \exists v_2 \rho(\mathbf{S}^a\mathbf{0}, v_2) \\ a \notin Q &\Rightarrow \langle a, b\rangle \notin R && \text{对于所有的 } b \\ &\Rightarrow A_E \vdash \neg\, \rho(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}) && \text{对于所有的 } b \\ &\Rightarrow A_E \nvdash \exists v_2 \rho(\mathbf{S}^a\mathbf{0}, v_2) \end{aligned}$$

最后一步的证明是正确的，因为如果对于所有的 b 都有 $A_E \vdash \neg\, \varphi(\mathbf{S}^b\mathbf{0})$，则 $A_E \nvdash \exists x \varphi(x)$. ($\omega$-和谐这个词就是指这个性质.) 由于对于 $\exists x\varphi(x), \neg\, \varphi(\mathbf{S}^0\mathbf{0}), \neg\, \varphi(\mathbf{S}^1\mathbf{0}), \cdots$，它们在 $\mathfrak{N}$ 中不可能都为真.

这样我们就有

$$a \in Q \Leftrightarrow A_E \vdash \exists v_2 \rho(\mathbf{S}^a\mathbf{0}, v_2).$$

要把这可表示性的一半归纳成定义并不困难.

定义 设 Q 是 $\mathbb{N}$ 上的 n 元关系, ψ 是一个公式, 其中只有 $v_1, \cdots, v_n$ 是自由变元. 则我们称 ψ 在理论 T 中弱表示 Q, 当且仅当对于 $\mathfrak{N}$ 中的每一组 $a_1, \cdots, a_n$,

$$\langle a_1, \cdots, a_n \rangle \in Q \Leftrightarrow \psi(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_n}\mathbf{0}) \in T.$$

观察到, 如果 Q 在和谐理论 T 中是可表示的, 则 Q 在 T 中也是可弱表示的.

定理 35J 一个关系在 Cn A_E 中是可弱表示的当且仅当它是递归可枚举的.

证明 我们只要证明一个递归可枚举的一元关系 Q 在 Cn A_E 中是可弱表示的, 那么同样的证明就可以运用到 n 元关系 Q 上, 而只需要改变一下符号就够了. 反之, 设 Q 由 Cn A_E 中的公式 ψ 弱表示, 则对于某个递归函数 f 和递归关系 P,

241

$$\begin{aligned} \langle a_1, \cdots, a_n \rangle \in Q &\Leftrightarrow \exists D\ [\ D\text{是}\psi(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_n}\mathbf{0})\text{从公理 } A_E \text{ 得到的推论}] \\ &\Leftrightarrow \exists d\ \langle\ d, f(a_1, \cdots, a_n) \rangle \in P \end{aligned}$$ ■

3.5.3 算术分层

我们定义一个自然数上的关系是 *算术的*, 当且仅当它在 $\mathfrak{N}$ 中是可定义的. 但从某种意义上说, 算术关系又比其他关系更容易定义. 我们可以根据定义关系的方法把算术关系归为一个分层.

设 Σ_1 为递归可枚举关系组成的类, 我们知道定义这些关系的表达式中总含有一个量词. 把这个想法拓展一下, 我们可以定义 Σ_k 关系类和 Π_k 关系类. 例如,

$$\Sigma_1: \{\vec{a} \mid \exists b \langle \vec{a}, b \rangle \in R\}, \text{R 是递归的}.$$

$$\Pi_1: \{\vec{a} \mid \forall b \langle \vec{a}, b \rangle \in R\}, \text{R 是递归的}.$$

$$\Sigma_2: \{\vec{a} \mid \exists c \forall b \langle \vec{a}, b, c \rangle \in R\}, \text{R 是递归的}.$$

$$\Pi_2: \{\vec{a} \mid \forall c \exists b \langle \vec{a}, b, c \rangle \in R\}, \text{R 是递归的}.$$

一般地, 关系 Q 在 Π_k 中当且仅当对于递归关系 R 它有形式

$$\{\vec{a} \mid \forall b_1 \exists b_2 \cdots \Box b_k \langle \vec{a}, \vec{b} \rangle \in R\}$$

如果 k 是奇数, 则 "□" 用 "∀" 代替, 如果 k 是偶数, 则 "□" 用 "∃" 代替. 类似地, Q 在 Σ_k 中当且仅当对于递归关系 R 它有形式

$$\{\vec{a} | \exists b_1 \forall b_2 \cdots \Box b_k \langle \vec{a}, \vec{b} \rangle \in R\}$$

如果 k 是奇数, 则 "□" 用 "∃" 代替, 如果 k 是偶数, 则 "□" 用 "∀" 来代替.

类 Σ_k 和 Π_k 也可以通过对 k 的递归来定义. Σ_1 是由递归可枚举关系组成的. 然后, 一个关系属于 Π_k 当且仅当它的补在 Σ_k 中. 一个关系属于 Σ_{k+1} 当且仅当它是 Π_k 中关系的定义域. (我们甚至可以从 $k = 0$ 开始, 通过令 Σ_0 为递归关系组成的类.)

例　由 A_E 数字确定的公式所对应的哥德尔数组成的集合在 Π_2 中.

证明　a 属于这个集合当且仅当 [a 是一个公式 α 的哥德尔数] 并且 $\forall b \exists d$[d 是从 A_E 形如 $\alpha(\mathbf{S}^{(b)_0}\mathbf{0}, \mathbf{S}^{(b)_1}\mathbf{0}, \cdots)$ 或其否定的一个推论的哥德尔数 $\mathcal{G}$]. 通过 3.4 节的技巧，我们可以证明方括号中句子定义的是递归关系. 我们将它们写成前束范式，就得到了所需要的形式，

242

$$\{a \mid \forall b \exists d \langle a, b, d \rangle \in R\},$$

且 R 是递归的. ■

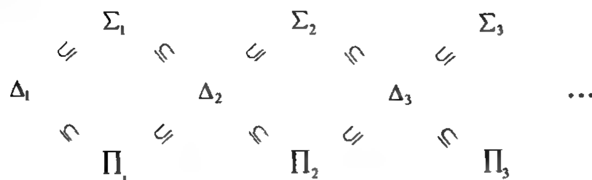
我们规定更多的记号：令 Δ_1 为递归关系组成的类，则前面的结果 (定理 35F) 陈述为，一个关系是递归的当且仅当它和它的补是递归可枚举的，可以用下面的等式来表示，

$$\Delta_1 = \Sigma_1 \cap \Pi_1.$$

由于这个等式的正确性，我们通过类似的等式来定义 $\Delta_n, n > 1$,

$$\Delta_n = \Sigma_n \cap \Pi_n.$$

下面的包含关系成立：



$\Delta_1 \subseteq \Sigma_1$ 的情况已经在前面提到过了 (参考定理 35F)，它的证明用到了"加入空量词"的方法. 从概念上讲，另一种情况的证明是相同的. 如果 φ 中不出现 x，则 φ, $\forall x \varphi$ 和 $\exists x \varphi$ 都是等价的. 比如说，Σ_1 中的一个关系由公式 $\exists y \varphi$ 定义，其中 φ 由 A_E 数字确定，那么这个关系同样可以由 $\exists y \forall x \varphi$ 和 $\forall x \exists y \varphi$ 来定义 (其中 x 不在 φ 中出现). 因此，这个关系也在 Σ_2 和 Π_2 中.

上述所有的包含关系都是真包含，即，等号不成立. 但在此我们不做证明. 我们把这些包含关系在图 3-1 中表示出来.

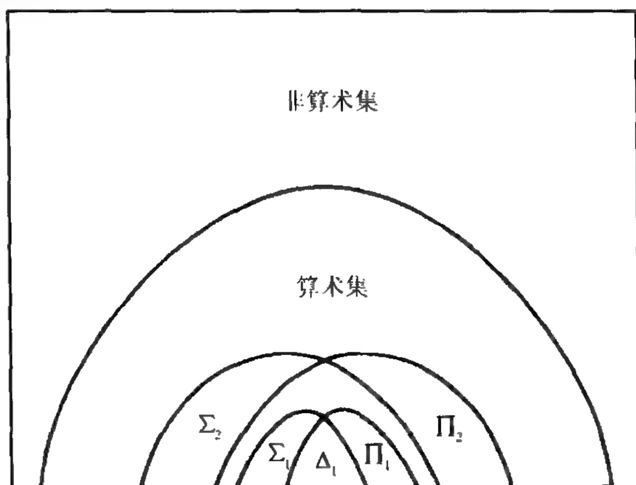


图 3-1　$\mathcal{PN}$ 的图像

算术关系组成的类等于 $\bigcup_k \Sigma_k$，也等于 $\bigcup_k \Pi_k$. 例如，任意 Σ_2 中的关系是算术的，它在 $\mathfrak{N}$ 中由公式 $\exists x \forall y \varphi$ 定义，这里 φ 由 A_E 数字确定. 反之，任意算术关系都可以在 $\mathfrak{N}$ 中由某个前束公式来定义. 这个前束公式中的无量词部分定义了一个递归关系 (因为无量词公式是由 A_E 数字确定的). 因此，被定义的关系就落入这个层次. 定理 35E 的证明中“依次去除”量词 $\exists\exists\cdots\exists$ 的方法 (及与之对称的是处理 $\forall\forall\cdots\forall$ 的方法) 在这里非常有用. 243

因此，我们有下面的结果，这个结果把 $\mathfrak{N}$ 中的可定义性和刚刚从递归关系中建立起来的层次联系了起来.

定理 35K 我们把自然数上的关系称为算术的 (即，在 $\mathfrak{N}$ 中是可定义的)，当且仅当对于某个 k 它在 Σ_k 中，并且这个条件等价于对于某个 l 它在 Π_l 中.

特别地，就像前面提到的那样，任意递归可枚举的关系是算术的.

当我们把一些特殊的算术关系进行分类时，一些技巧是非常有用的. 例如，设 A 是一些公式 α 的哥德尔数集，使得对于某个 n，

$$A_E \vdash \alpha(\mathbf{S}^n\mathbf{0}) \text{ 且 } (\forall i < n) A_E \vdash \neg\, \alpha(\mathbf{S}^i\mathbf{0}).$$

那么 $a \in A$ 当且仅当 [a 是一个合式公式 α 的哥德尔数] 并且 $\exists n \exists D$[D 是从 A_E 得到的形如 $\alpha(\mathbf{S}^n\mathbf{0})$ 的推论] 并且 $(\forall i < n)\exists D_i$[$D_i$ 是从 A_E 得到的形如 $\neg\, \alpha(\mathbf{S}^i\mathbf{0})$ 的推论]. 方括号内的部分是递归的，因此我们要数一数剩下的量词. 约束量词“$\forall i < n$”就不必数了，因为我们有

$$(\forall i < n)(\exists d)\langle d, i\rangle \in P \Leftrightarrow (\exists d)(\forall i < n)\langle (d)_i, i\rangle \in P.$$

利用这个事实，我们把约束量词推到递归部分中. 这样就有，$A \in \Sigma_1$.

下面的定理是定理 35I 的推广. 244

定理 35L 设 A 是句子的集合，使得 $\sharp A$ 在 Σ_k 中，这里 $k > 0$，则 $\sharp \mathrm{Cn}\ A$ 也在 Σk 中.

证明 回到 3.4 节中的 18 和 20 条的证明. 在这里我们有：

$a \in \sharp \mathrm{Cn}\ A \Leftrightarrow a$ 是一个句子的哥德尔数且 $\exists d$[d 是数字序列并且 d 的最后部分是 a 并且对于每个小于 $\mathrm{lh} d$ 的 i，要么 (1)$(d)_i \in \sharp\ A$，(2)$(d)_i$ 是逻辑公理的哥德尔数，要么 (3) 对于小于 i 的某个 j 和 l，$(d)_j = \langle h(()\rangle * (d)_l * \langle h(\rightarrow)\rangle * (d)_i * \langle h())\rangle$].

因为在 (1) 中 $\sharp A \in \Sigma_k$，我们必须把“$(d)_i \in \sharp A$”对递归 Q 用下面的形式替换：

$$\exists b_1 \forall b_2 \cdots \Box b_k \langle (d)_i, \vec{b}\rangle \in Q \qquad \blacksquare$$

现在只剩下把这个结果转化成 Σ_k 中的自然语言前束表达式. 我们假定读者设 $k = 2$ 并且已经写出了表达式，上个例子中的过程对此将有帮助.

习题

1. 证明不存在递归集 R 使得 $\sharp \mathrm{Cn}\ A \subseteq R$ 并且 $\sharp\{\sigma | (\neg\ \sigma) \in \mathrm{Cn}\ A_E\} \subseteq \bar{R}$，$\bar{R}$ 是 R 的补. (这个结果可以叙述为：A_E 的定理不能和可拒绝的句子递归地分开.) 提示：构造一个句子 σ“我的哥德尔数不在 R 中.”检验一下 $\sharp\sigma$ 在哪儿.
2. 设在含有 $\mathbf{0}$ 和 $\mathbf{S}$ 的递归可枚举语言中，A 是句子的递归集合. 同时假设每个递归关系在理论 $\mathrm{Cn}\ A$ 中都是可表示的. 进一步假设 A 是 ω 和谐的，即，不存在公式 φ 使得 $A \vdash x \exists x \varphi(x)$ 而且对于所有的 $a \in \mathbb{N}$，$A \vdash \neg\ \varphi(\mathbf{S}^a\mathbf{0})$. 构造一个句子 σ，σ 间接表示它不是 A 的定理，并证明既不是 $A \vdash \sigma$，也不是 $A \vdash \neg\ \sigma$. 提示：参考 3.0 节.

说明：这是不完全性定理的另一种表达形式，它和哥德尔 1931 年提出的原始定理很相近．请注意，我们不要求公理 A 在 $\mathfrak{N}$ 中是真的，也不要求 A 包含 A_E．但仍然可以使用不动点的讨论．

3. 令 T 是递归可数语言 (包含 **0** 和 **S**) 中的理论．假设 $\mathbb{N}$ 的所有递归子集在 T 中都可弱表示．证明：$\sharp T$ 不是递归的．*提示*：构造一个二元关系 P 使得 $\mathbb{N}$ 的任意可弱表示的子集都等于 $\{b|\langle a,b\rangle \in P\}$，其中 a 是某个元素，并且满足如果 $\sharp T$ 是递归的，则 P 也是递归的．考虑集合 $H=\{b|\langle b,b\rangle \notin P\}$，参考 3.0 节中的证明方法．在那一节中，针对特殊情况 $T=\mathrm{Th}\mathfrak{N}$ 所使用的"对角线法"在这里可以采用．

245

说明：这个练习给出了下面这个结论，"任意一个足够强的理论都是不可判定的．"

4. 证明存在 $2^{\aleph_0}$ 个互不同构的 $\mathrm{Th}\,\mathfrak{N}$ 的可数模型．*提示*：对于每个素数集 A，构造一个模型，使得模型中有一个元素恰好能被 A 中的素数整除．
5. (Lindenbaum) 设 T 是可判定的和谐理论 (在一个合理的语言中)．证明 T 可以被扩展成可判定的完全和谐理论 T'．*提示*：依次检验每个句子 σ；向 T 中添加 σ 或 $\neg\sigma$．但要注意保持可判定性．
6. 我们考虑等号的语言，其中 $\forall$ 是唯一的参数．令 λ_n 是"至少存在 n 个元素"的公式表示，参见定理 26 A 的证明．我们称一个公式是 *简单的* 当且仅当它可以从原子公式建立起来，并且 λ_n 是通过使用联结符号得到的(而不是使用量词符号)．现给定等号的语言中的任意一个公式，请读者给出找到与其逻辑等价的简单公式的方法．*提示*：把这个习题看作是量词消去的结果 (其中在 λ_n 中量词不出现)．运用定理 31F．
7. (a) 假设 A 和 B 是属于 Σ_k(或 Π_k) 的 $\mathbb{N}$ 的子集．证明 $A\cup B$ 和 $A\cap B$ 也属于 Σ_k(或 Π_k)．

   (b) 假设 A 在 Σ_k(或 Π_k) 中，并且函数 $f_1,\cdots,f_m$ 是递归的．证明

$$\{\vec{a}:\langle f_1(\vec{a}),\cdots,f_m(\vec{a})\rangle\in A\}$$

   也在 Σ_k(或 Π_k) 中．*提示*：首先证明该结论对 Σ_1 成立，然后可以把讨论的过程推广．
8. 令 T 是递归可枚举语言 (含有 **0** 和 **S**) 中的理论，令 $n\geqslant 0$ 是一个固定的数．假设所有在 Σ_n 中的 $\mathbb{N}$ 的子集都可在 T 中弱表示．证明 $\sharp T$ 不在 Π_n 中．(习题 3 是本题当 $n=0$ 时的特例，只需把当时的提示用在现在的情况就可以了．)
9. 证明

$$\{\sharp\sigma|A_E;\sigma\text{是 }\omega\text{ 和谐的}\}$$

   (见习题 2) 是 Π_3 集合．
10. 理论 $\mathrm{Cn}\,A_E$ 有许多完全的扩展，$\mathrm{Th}\,\mathfrak{N}$ 仅仅是其中的一个，那么像这样的完全扩展有多少呢？即，A_E 的完全扩展理论 (在语言中) 所组成的集合的基数是多少？

246

3.6　递归函数

为了得到不完全性定理和不可判定性定理，我们已经使用了递归函数 (即，那些被看作关系的函数是递归的)．就其本身而言，递归函数类也是一个有趣的类，在本节中，我们将介绍一些它的性质．

根据丘奇论题，我们已经知道，一个函数是递归的当且仅当它能够由一个能行过程来计算．这个结果在递归函数中非常有用，并且能够帮助我们直观地理解递归这个概念，进而有利于我们对递归函数的学习．例如，假设已知 $\mathbb{N}$ 的一个递归置换，问它的逆是否是递归的？在证明之前，我们首先应该直观地考虑一下它所对应的问题：一个可计算的置换 f 的反函数是否是可计算的？答案是肯定的．要计算 $f^{-1}(3)$，我们可以分别计算 $f(0),f(1),\cdots$ 直到找到

某个 k 使得 $f(k) = 3$. 那么 $f^{-1}(3) = k$. 至此，我们可以得到两个结果. 第一，关于递归置换的答案也是肯定的. 第二，证明这个结论的方法，只需将直观证明严格地写出来就行了. 在本节中，上述这个解决有关递归问题的方法是非常有用的.

在开始讨论递归函数之前，先列出一些有关递归函数的事实. 根据定理 34A 我们知道，一个函数 f 是递归的当且仅当它 (作为关系) 在 $\mathrm{Cn}A_E$ 中可表示. 进而，每一个递归函数在这个理论中是可弱表示的.

3.3 节中给出了递归函数的编目表，并且证明了递归函数类在某些运算下，比如复合运算 (定理 33L) 和 "最小零" 运算 (定理 33M)，是封闭的.

我们还知道一些函数不是递归的. 虽然从 $\mathbb{N}^m$ 到 $\mathbb{N}$ 内的函数有不可数多个 (精确地说有 $2^{\aleph_0}$ 个)，但其中只有可数多个是递归的. 尽管在 3.3 节中提到的函数几乎都是递归的，但仍然存在着大量不可递归的函数. 从 3.3 节的编目 1 可知，一个非递归集合的特征函数不是递归的. 例如，如果当 a 是 $\mathrm{Cn}A_E$ 中的元素的哥德尔数时，$f(a) = 1$，否则 $f(a) = 0$，那么 f 就不是递归的.

247

3.6.1 范式

对于任意一个可计算的函数，譬如多项式函数 $a^2 + 3a + 5$，我们原则上可以设计一个数字计算机，使得当我们输入 a 时，计算机能输出 $a^2 + 3a + 5$ 的值 (见图 3-2). 但如果我们要计算另外一个不同的函数，就要设计另一个不同的计算机. (或改变原有计算机的线路.) 很早以前人们就意识到，需要设计一种具有内装通用程序的计算机. 使用这种计算机时，只要输入 a 和计算多项式所需要的程序 (见图 3-3) 就可以. 这种 "通用" 计算机需要两个输入项，并且只要输入的计算程序是正确的，(如果内存空间足够大) 它就可以计算任意的一元可计算函数. 当然，许多程序员也发现，有一些程序不和 $\mathbb{N}$ 上的任意一个函数所对应. (它们实际上会产生故障!)

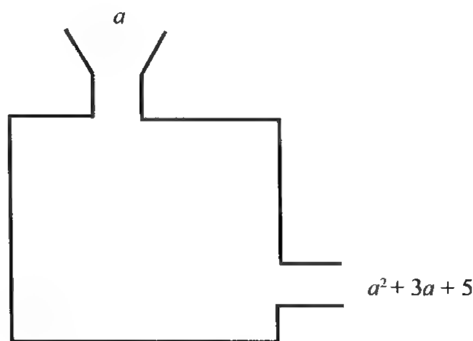


图 3-2 特殊用途的计算机

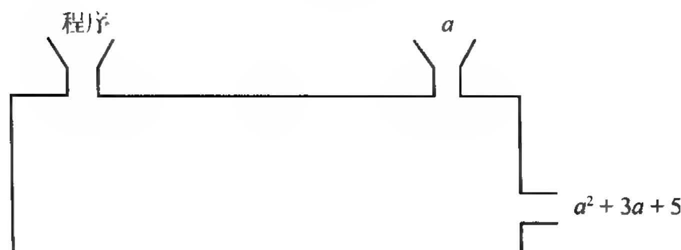


图 3-3 通用计算机

在这一节和下一节中，将重复上面的内容，但用的是递归函数及其证明. 对于通用计算机，我们有递归关系 T_1 和递归函数 U. 那么对于任意一个递归函数 $f:\mathbb{N}\to\mathbb{N}$，存在一个 e(类似于程序) 使得

$$\begin{aligned} f(a) &= U(\text{使得}\langle e,a,k\rangle\in T_1\text{的最小的 }k) \\ &= U(\mu k\langle e,a,k\rangle\in T_1), \end{aligned}$$

248 其中第二个式子可以看作第一个的缩写. 实际上，这里的 e 是公式 φ 的哥德尔数，φ 在 $\mathrm{Cn}A_E$ 中表示 (至少弱表示)f. 使得 $\langle e,a,k\rangle\in T_1$ 的数 k 编码从 $\varphi(\mathbf{S}^a\mathbf{0},\mathbf{S}^{f(a)}\mathbf{0})$ 的 A_E 推理的哥德尔数 $\mathcal{G}$ 与 $f(a)$.

定义　对于每个正数 m，令 T_m 为 $(m+2)$ 元关系，使得 $\langle e,a_1,\cdots,a_m,k\rangle\in T_m$ 当且仅当

(i) e 是自由变元为 $v_1,\cdots,v_m,v_{m+1}$ 的公式 φ 的哥德尔数；

(ii) k 是长度为 2 的数字序列，并且 $(k)_0$ 是从 $\varphi(\mathbf{S}^{a_1}\mathbf{0},\cdots,\mathbf{S}^{a_m}\mathbf{0},\mathbf{S}^{(k)_1}\mathbf{0})$ 的 A_E 推出的哥德尔数 $\mathcal{G}$.

这个定义的思想是，对于任意的一元递归函数 f，我们首先把 e 取为弱表示 f(看作关系) 的公式 φ 的哥德尔数. 那么我们知道，对于任意 a 和 b,

$$A_E\vdash\varphi(\mathbf{S}^a\mathbf{0},\mathbf{S}^b\mathbf{0})\quad\text{iff}\quad b=f(a)$$

所以满足定义中的条件 (ii) 的任意数 k 必须等于 $\langle(k)_0,f(a)\rangle$，其中 $(k)_0$ 是来自 A_E 的 $\varphi(\mathbf{S}^a\mathbf{0},\mathbf{S}^{f(a)}\ \mathbf{0})$ 的推理的 $\mathcal{G}$. (在此，不要求 T_m 定义中的 k 尽量小，因此可以把它从 T_m 通常的定义中分出.)

我们把 "结果" 函数 U 取成

$$U(k)=(k)_1.$$

则 U 是递归的，并且根据上一段中的条件，我们有 $U(k)=f(a)$.

推论 36A　对于每个 m，关系 T_m 是递归的.

证明　$m=2$ 的情况. $\langle e,a_1,a_2,k\rangle\in T_2$ 当且仅当 e 是一个公式的哥德尔数，$\sharp(\forall v_1\forall v_2\forall v_3)*e$ 是一个句子的哥德尔数，k 是长度为 2 的数字序列，并且 $(k)_0$ 是从 Sb (Sb (Sb $(e,\sharp v_1,g(a_1))$, $\sharp v_2$, $g(a_2))$, $\sharp v_3$,$g((k)_1))$ 的 A_E 推理的哥德尔数 $\mathcal{G}$，其中 $g(n)=\sharp\mathbf{S}^n\mathbf{0}$. 从 3.4 节中我们知道，所有这些都是递归的. ■

定理 36B　(a) 对于任意的递归函数 $f:\mathbb{N}^m\to\mathbb{N}$，存在一个 e 使得对于所有 $a_1,\cdots,a_m$,

$$f(a_1,\cdots,a_m)=U(\mu k\langle e,a_1,\cdots,a_m,k\rangle\in T_m).$$

(特别地，这样的数 k 是存在的.)

(b) 反之，对于使得 $\forall a_1,\cdots,a_m\exists k\langle e,a_1,\cdots,a_m,k\rangle\in T_m$ 的 e，在 $a_1,\cdots,a_m$ 的值为

249 $U(\mu k\ \langle e,a_1,\cdots,a_m,k\rangle\in T_m)$ 的函数是递归的.

证明 (b) 可以从 U 和 T_m 是递归的事实得出. 至于 (a), 我们把 e 取成在 $\text{Cn}A_E$ 中弱表示 f 的公式 φ 的哥德尔数. 对于任意给定的 $\vec{a}$, 我们知道 $A_E \vdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{f(\vec{a})}\mathbf{0})$. 如果我们令 d 是从这个句子的 A_E 推出的 $\mathcal{G}$, 那么 $\langle e, \vec{a}, \langle d, f(\vec{a})\rangle\rangle \in T_m$. 因此存在某个 k, 使得 $\langle e, \vec{a}, k\rangle \in T_m$. 并且对于任意这样的 k, 我们知道由于 $(k)_0$ 是 $\mathcal{G}$ 的推论, 所以 $A_E \vdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^{(k)_1}\mathbf{0})$. 进而, 根据我们对 φ 的选择有 $U(k) = (k)_1 = f(\vec{a})$. 这样, 我们有 $U(\mu k\langle e, \vec{a}, k\rangle \in T_m) = f(\vec{a})$ ■

这个定理是 Kleene 在 1936 年提出的, 它说明了每一个递归函数在范式

$$f(\vec{a}) = U(\mu k\langle e, \vec{a}, k\rangle \in T_m).$$

的形式下是可表示的. 这样, 能够计算 U 和 T_1 的特征函数的计算机就是一元递归函数的"通用"计算机. 输入项 e 对应于这个程序, 如果输出项是任意的, (即, 对于任意的 k 都有 $\langle e, a, k\rangle \in T_1$.) 那么 e 的选择必须很小心.

3.6.2 部分递归函数

如果我们把部分函数也考虑进来, 递归函数的理论就变得更自然了.

定义 m 元 *部分函数* 是指定义域 $\text{dom}\, f \subseteq \mathbb{N}^m$ 且值域 $\text{ran}\, f \subseteq \mathbb{N}$ 的函数 f. 如果 $\vec{a} \notin \text{dom}\, f$, 则称 $f(\vec{a})$*无定义*. 如果 $\text{dom}\, f = \mathbb{N}^m$, 则称 f 是 *全函数*.

在此提醒大家不要过于关注"部分"和"全体"这两个词 (或"无定义"). 我们说部分函数 f 可以是也可以不是全体的. "部分"和"全体"这两个词不是对立的.

首先来看那些不是正规意义上可计算的部分函数.

定义 m 元部分函数 f 是 *可计算的* 当且仅当存在一个可行的过程使得

(a) 对于 $\text{dom}\, f$ 中的 m 元组 $\vec{a}$, 这个过程能够得出 $f(\vec{a})$; 并且

(b) 对于不在 $\text{dom}\, f$ 中的 $\vec{a}$, 这个过程没有输出值.

这个定义对全函数的可计算性做出了扩展. 在 3.3 节中我们证明了定理 33H, 这个定理中的一些结果对部分函数仍然成立.

***定理 36C** m 元部分函数 f 是可计算的当且仅当 f(作为 $(m+1)$ 元关系) 是能行可枚举的.

250

证明 该定理的证明与定理 17E 的证明很相似. 首先假设我们能够能行地枚举 f. 对于给定的 m 元组 $\vec{a}$, 我们逐个检查枚举过程列出的关系. 如果一个以 $\vec{a}$ 开头的 $(m+1)$ 元组出现了, 我们就将这个 $(m+1)$ 元组的最后一个元素作为 $f(\vec{a})$ 的值.

另一方面, 假设 f 是可计算的, 并且 f 是个一元部分函数. 我们可以按照下面的步骤把 f 当作关系枚举出来:

(1) 花一分钟来计算 $f(0)$.

(2) 花两分钟来计算 $f(0)$, 再花两分钟计算 $f(1)$.

(3) 花三分钟来计算 $f(0)$, 花三分钟计算 $f(1)$, 再花三分钟计算 $f(2)$.

如此计算下去, 当然, 当其中一步算出结果时, 我们就把对应的一对变量和结果作为关系 f 的一个元素.

对于可计算的 m 元部分函数，我们计算的不是 f 在 $0,1,2,\cdots$ 的值，而是它在 $\langle(0)_0,\cdots,(0)_{m-1}\rangle,\langle(1)_0,\cdots,(1)_{m-1}\rangle,\langle(2)_0,\cdots,(2)_{m-1}\rangle$ 等等上的值. ■

当 f 是可计算的全函数时，我们也能得到 f 是一个可判定的关系. 但对于部分函数来说，这个结果不成立. 例如，令

$$f(a)=\begin{cases}0 & \text{如果}a\in\sharp\mathrm{Cn}A_E,\\ \text{无定义} & \text{其他情况}.\end{cases}$$

则 f 是可计算的. (我们通过枚举 $\sharp\mathrm{Cn}A_E$ 并找到 a 来计算 $f(a)$.) 但 f 不是一个可判定关系，否则 $\sharp\mathrm{Cn}A_E$ 就是可判定的了. 根据这个例子和上一个定理，我们给出与可计算部分函数对应的部分递归函数的确切定义.

定义　如果一个部分函数在被看作关系时是递归可枚举的，则称这个函数为部分递归函数.

大家要注意的是，"部分递归函数" 是一个不能再分割的短语；即一个部分递归函数 (作为关系) 不必是递归的. 但对于全函数来说，这个术语中的 "递归" 仍然和前面的意思相同.

定理 36D　令 $f:\mathbb{N}^m\to\mathbb{N}$ 是一个全函数，则 f 是部分递归函数当且仅当 f 是递归的 (作为关系).

251

证明　如果 f(作为关系) 是递归的，那么 f 更是递归可枚举的. 反之，假设 f是递归可枚举的. 由于 f 是全函数，所以

$$f(\vec{a})\neq b\Leftrightarrow\exists c[f(\vec{a})=c\text{且}b\neq c].$$

等价号右侧的表达式说明 f 的补集也是递归可枚举的. 这样根据定理 35F，f 是递归的. ■

在范式部分的讨论中，我们画了一个双输入的装置 (见图 3-4). 对于任意一个可计算的部分函数，应该存在某个程序能计算它. 但目前这个断言的逆命题成立：任意一个程序都能生成某个可计算的部分函数. 当然，许多程序生成的函数是空的，但空函数仍然是可计算的部分函数.

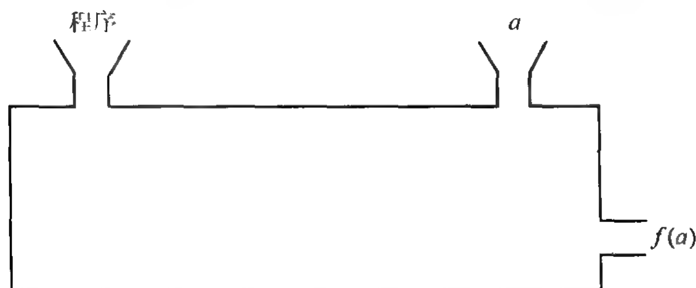


图 3-4　带有计算 f 的程序的计算机

我们可以用相同的方法来考虑部分递归函数. 对于每个 $e\in\mathbb{N}$，我们定义 m 元部分函数 $[\![e]\!]_m$ 为

$$[\![e]\!]_m(a_1,\cdots,a_m)=U(\mu k\langle e,a_1,\cdots,a_m,k\rangle\in T_m).$$

如果这样的 k 不存在，等号的右边可以理解为无定义．换句话说，

$$\vec{a} \in \operatorname{dom} [\![e]\!]_m \quad \text{iff} \quad \exists k \langle e, a_1, \cdots, a_m, k \rangle \in T_m,$$

在这种情况下，$[\![e]\!]_m(\vec{a})$ 的值由上面的等式给出．

下面的定理是对定理 36B 的推广．

范式定理 (Kleene, 1943) (a) 在 $\langle e, a_1, \cdots, a_m \rangle$ 的值是 $[\![e]\!]_m(a_1, \cdots, a_m)$ 的 $(m+1)$ 元部分函数是部分递归函数．

(b) 对于每个 $e \geqslant 0$，$[\![e]\!]_m$ 是 m 元部分递归函数．

(c) 对于任意 m 元部分递归函数，都存在某个 e，使得这个部分递归函数等于 $[\![e]\!]_m$．

证明 (a) 已经有

$$[\![e]\!]_m(\vec{a}) = b \Leftrightarrow \exists k[\langle e, \vec{a}, k \rangle \in T_m,\ U(k) = b \text{且} (\forall k' < k) \langle e, \vec{a}, k' \rangle \notin T_m].$$

252

方括号中的部分是递归的，因此这个函数 (作为关系) 是递归可枚举的．

(b) 的证明仍然可以采用 (a) 中的方法，只不过此时 e 是固定的．

(c) 令 f 是一个 m 元部分递归函数，则 $\{\langle \vec{a}, b \rangle, | f(\vec{a}) = b\}$ 是递归可枚举的．因此存在公式 φ 在 $\mathrm{Cn}A_E$ 中弱表示这个关系．我们断言 $f = [\![\sharp\varphi]\!]_m$．因为如果 $f(\vec{a}) = b$，则 $A_E \vdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^b\mathbf{0})$．所以存在 k 使得 $\langle \sharp\varphi, \vec{a}, k \rangle \in T_m$．对于任意这样的 k，我们有 $U(k) = b$，这是因为对于 $c \neq b$，$A_E \nvdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^c\mathbf{0})$．类似地，如果 $f(\vec{a})$ 无定义，则对于任意的 c，$A_E \nvdash \varphi(\mathbf{S}^{a_1}\mathbf{0}, \cdots, \mathbf{S}^{a_m}\mathbf{0}, \mathbf{S}^c\mathbf{0})$，因此 $[\![\sharp\varphi]\!]_m$ 在这也无定义． ■

范式定理中的 (a)(当 $m = 1$ 时) 告诉我们，由等式

$$\Phi(e, a) = [\![e]\!]_1(a) = U(\mu k \langle e, a, k \rangle \in T_1)$$

定义的函数 Φ 是部分递归函数．同时 (c) 说明，如果我们固定 Φ 的第一个变量于一个合适的值，我们可以得到任意的一元部分递归函数．从这个意义上说，Φ 是“通用的”．

与通用函数相对应的非正规概念是计算机的操作系统．操作系统需要两个输入：程序 e 和数据 a．并利用数据运行程序．但操作系统本身，和一个二元部分函数一样，是 可计算的．

范式定理的证明为我们提供了一种方法来计算“操作系统” Φ 的值，虽然这种方法的效率非常低．至少“根据程序 e 对数据 a 执行操作”这一直接的思想没有得到体现．

函数 $[\![e]\!]_m$ 被称为以 e 为 指标 的 m 元部分递归函数．范式定理中的 (c) 告诉我们每个递归部分函数都有一个指标．它的证明过程说明，弱表示这个函数的公式所对应的哥德尔数往往是该函数的指标．

对于所有的一元部分递归函数，现在有了方便的标记方法 $[\![0]\!]_1, [\![1]\!]_1, \cdots$．函数 $[\![e]\!]_1$ 是由 e 所编码的“指令”产生的．当然，如果 e 不是公式的哥德尔数，或其他的情况没有得到满足，那么对应的函数将是空的．

在部分递归函数的枚举中包括了所有的递归全函数．但我们不能能行地判断一个数 e 是不是某个全函数的指标．

定理 36E 集合 $\{e | [\![e]\!]_1$ 是全函数 $\}$ 不是递归的．

证明　我们把这个集合记为 A. 考虑下面式子所定义的函数:

$$f(a)=\begin{cases}[\![a]\!]_1(a)+1 & \text{如果}a\in A,\\ 0 & \text{如果}a\notin A.\end{cases}$$

253　根据 f 的构造, 它是全函数. 它是不是递归的? 有

$$f(a)=b\Leftrightarrow[(a\notin A\text{且}b=0)\text{或}(a\in A\ ,\ \exists k(\langle a,a,k\rangle\in T_1\ ,\\ b=U(k)+1\text{ 且 }(\forall j<k)\langle a,a,j\rangle\notin T_1))].$$

这样, 如果 A 是递归的, 那么 f(作为关系) 是递归可枚举的. 又如果 f 是个全递归函数, 则它等于 $[\![e]\!]_1$, $e\in A$. 但由于 $f(e)=[\![e]\!]_1(e)+1$, 所以我们不能得到 $f=[\![e]\!]_1$. 这个矛盾证明了 A 不可能是递归的. ■

不难证明 A 属于 Π_2. 这个分类不可能是最好的, 因为可以证明 A 不属于 Σ_2.

定理 36F　集合 $K=\{a|[\![a]\!]_1(a)$ 是可定义的 $\}$ 是递归可枚举的, 但不是递归的.

证明　由于 $a\in K\Leftrightarrow\exists k\langle a,a,k\rangle\in T_1$, 所以 K 是递归可枚举的. 为了证明 K 不是递归的, 考虑如下定义的函数:

$$g(a)=\begin{cases}[\![a]\!]_1(a)+1 & \text{如果}a\in K,\\ 0 & \text{如果}a\notin K.\end{cases}$$ ■

这是一个全函数. 和上个定理完全一样, 我们有 K 不可能是递归的.

推论 36G(停机问题的不可解性)　关系 $\{\langle e,a\rangle|[\![e]\!]_1(a)$ 是可定义的 $\}$ 不是递归的.

证明　我们已知 $a\in K$ 当且仅当 $\langle a,a\rangle$ 属于这个关系. (这样 K 中元素的问题就可"简化"为停机问题.) 如果这个关系是递归的, 那么 K 也是递归的, 但实情并不是这样. ■

这个推论告诉我们, 对于一个部分递归函数的程序 e 和输入 a, 不存在一个能行的方法来判断函数 $[\![e]\!]_1$ 在 a 上是否有定义.

我们可以通过下面的特征得到递归可枚举关系的指标.

定理 36H　$\mathbb{N}$ 上的关系是递归可枚举的当且仅当它是某个部分递归函数的定义域.

254　**证明**　任意递归可枚举关系的定义域也是递归可枚举的, 参见定理 35E 的第 4 部分. 特别地, 任意部分递归函数的定义域也是递归可枚举的.

反之, 令 Q 是任意的递归可枚举关系, 其中

$$\vec{a}\in Q\Leftrightarrow\exists b\langle\vec{a},b\rangle\in R$$

且 R 是递归的. 设

$$f(\vec{a})=\mu b\langle\vec{a},b\rangle\in R;$$

即

$$f(\vec{a})=b\Leftrightarrow\langle\vec{a},b\rangle\in R\text{ 且 }(\forall c<b)\langle\vec{a},c\rangle\notin R.$$

那么, f 作为一个关系是递归的. 因此 f 是部分递归函数. 显然它的定义域是 Q. ■

这样，部分递归函数的指标就可以诱导出递归可枚举关系的指标. 定义

$$W_e = \mathrm{dom}[\![e]\!]_1.$$

则 $W_0, W_1, W_2, \cdots$ 就列举出了 $\mathbb{N}$ 的所有递归可枚举子集. 在定理 36E 中，我们证明了 $\{e|W_e = \mathbb{N}\}$ 不是递归的. 类似地，定理 36F 断言 $\{e|e \in W_e\}$ 也不是递归的. 我们定义一个关系 Q:

$$Q = \{\langle e, a\rangle | a \in W_e\}.$$

那么 Q 是递归可枚举的，这是因为 $\langle e, a\rangle \in Q \Leftrightarrow \exists k \langle e, a, k\rangle \in T_1$. 进而，对于任意递归可枚举集 $A \subseteq \mathbb{N}$，存在某个 e，使得 $A = \{a | \langle e, a\rangle \in Q\}$，从这个意义上说，$Q$ 对于递归可枚举集是通用的. 不可解决的停机问题可以描述为：Q 不是递归的.

我们可以用经典的对角线法把递归可枚举集 $W_0, W_1, W_2, \cdots$ 对角线化. 那么集合 $\{a|a \notin W_a\}$ 不等于任意一个 W_q. 实际上，这个集合正好是定理 36F 中集合 K 的补集 $\overline{K}$. 因为

$$q \in \overline{K} \Leftrightarrow q \notin W_q,$$

集合 $\overline{K}$ 不可能等于 W_q，数 q 就是 $\overline{K}$ 和 W_q 不相等的证据.

不仅如此，只要 W_q 是 $\overline{K}$ 的递归可枚举子集，即 $W_q \subseteq \overline{K}$，我们就能在 $\overline{K}$ 中找到一个不在 W_q 中的数字. 这个数字就是 q 本身. 为了证明这一点，我们回到上一段中的等价条件，由于 $W_q \subseteq \overline{K}$，等价号两边不可能都不成立 ($q \in K$ 和 $q \in W_q$). 因此两边都是对的.

定理 36F 说明，尽管 K 是递归可枚举的，但它不是递归的. 为了证明它不是递归的，只需要证明它的补集 $\overline{K}$ 不是递归可枚举的. 但上面的证明用了更强的一种方法，因而为我们证明定理 36F 提供了另一种途径. [255]

现在，让我们从可计算性的角度来重新考虑哥德尔不完全性定理.

由于集合 K 是递归可枚举的 (即，Σ_1)，根据定理 35K，K 是算术的，也即 K 在结构 $\mathfrak{N}$ 中是可定义的.

所以存在只含有自由变元 v_1 的公式 $\varkappa(v_1)$ 在 $\mathfrak{N}$ 中定义了 K. 进而公式 $\neg\varkappa(v_1)$ 在 $\mathfrak{N}$ 中定义了 $\overline{K}$. 这样，我们有

$$a \in \overline{K} \Leftrightarrow (\neg\varkappa(\mathbf{S}^a\mathbf{0})) \in \mathrm{Th}\mathfrak{N}.$$

这个结果将判断集合 $\overline{K}$ 中元素的问题归结为有关 Th$\mathfrak{N}$ 的问题. 假定已知数 a，我们想知道 a 是否属于 $\overline{K}$. 我们可以 计算$\sharp(\neg\varkappa(\mathbf{S}^a\mathbf{0}))$. (通俗地说，我们显然可以 能行的 计算 $\sharp(\neg\varkappa(\mathbf{S}^a\mathbf{0}))$. 正式地说，3.4 节中的第 5 条确保我们可以递归地计算出这个数.) 假想我们已经知道了 $\sharp$Th $\mathfrak{N}$ 的“预言”(即一个假想的装置，输入一个数，它就能判断这个数是否在 $\sharp$Th $\mathfrak{N}$ 中)，那么我们能够回答“是否 $a \in \overline{K}$?”

现在让我们抛弃幻想. 对于自然数集合 A 和 B，我们称 A 多一可归约于 B(用符号表示为 $A \leqslant_m B$) 当且仅当存在一个全递归函数 f 使得对于每个数 a，

$$a \in A \Leftrightarrow f(a) \in B.$$

前一个例子告诉我们 $\overline{K} \leqslant_m \sharp\mathrm{Th}\mathfrak{N}$. 更一般地说，前面的讨论说明了任意一个算术集合多一可归约于 $\sharp\mathrm{Th}\mathfrak{N}$.

引理 36I　假设 A 和 B 是自然数集合并且满足 $A \leqslant_m B$.

(a) 如果 B 是递归的，那么 A 也是递归的.

(b) 如果 B 是递归可枚举的，那么 A 也是递归可枚举的.

(c) 如果 B 属于某个 Σ_n，那么 A 也属于这个 Σ_n.

证明　(a) 是我们很熟悉的，3.3 节中的编目 2 就是用另一个方法表达的 (a).

(b) 实质上是 (a)"加一个量词"后的结果. 也就是，由于 B 是递归可枚举的，我们知道对于某个递归的二元关系 Q，

$$c \in B \Leftrightarrow \exists b Q(c,b).$$

如果 f 是全递归函数，使得 A 多一可归约于 B，那么对于每个数 a，

$$a \in A \Leftrightarrow f(a) \in B \Leftrightarrow \exists b[Q(f(a),b)].$$

256 和引理中的 (a) 一样，方括号内的部分是递归的 (即，$\{\langle a,b\rangle | Q(f(a),b)\}$ 是递归的). 因此 A 是递归可枚举的.

(c) 实质上是 (a)"加 n 个量词"后的结果. 证明过程和 (b) 类似. ■

我们检验 $\overline{K}$ 这个特殊的集合的原因是它能够给出下列的结果：

哥德尔不完全性定理　$\mathrm{Th}\mathfrak{N}$ 不能递归公理化.

证明　$\mathrm{Th}\mathfrak{N}$ 不能递归公理化，根据前一个引理，否则 $\overline{K}$ 是递归可枚举的. 但是任意一个递归可公理化的理论必然是递归可枚举的 (3.4 节中的第 20 条，也见定理 35I). ■

简单地说，证明过程就是：任意递归可公理化的理论是递归可枚举的，但 $\mathrm{Th}\mathfrak{N}$ 不是递归可枚举的. 因此任意递归可公理化的子理论一定是不完全的.

我们有必要复习一下这个证明，但可以用肯定的结论来代替否定的结论 (什么样什么样的集合没有某种特殊的性质).

假设 T 是 $\mathrm{Th}\mathfrak{N}$ 的任意一个递归可公理化的子理论. (因此根据上面的定理，T 是不完全的.) 我们希望找到一个句子来证明它的不完全性.

我们已经有了一个全递归函数 f，它使得 $\overline{K}$ 多一可归约于 $\sharp\mathrm{Th}\mathfrak{N}$，也就是 $f(a) = \sharp(\neg\chi(\mathbf{S}^a\mathbf{0}))$，则对于每个 a，

$$a \in \overline{K} \Leftrightarrow f(a) \in \sharp\mathrm{Th}\mathfrak{N}.$$

那么 $f(a)$ 就是句子"$a \notin K$"(的哥德尔数).

考虑由下列条件定义的数集 J

$$a \in J \Leftrightarrow f(a) \in \sharp T.$$

那么 J 由那些不在 K 中的数组成，并且 T"知道"它们不在 K 中. 我们考察 J 的两个方面：

首先，J 是递归可枚举的. J 由 f 多一归约于 $\sharp T$，再利用引理 36I(b) 而得.

其次，$J \subseteq \overline{K}$. 我们已知 $T \subseteq \mathrm{Th}\mathfrak{N}$，所以如果从 T 能得到 $a \notin K$，则肯定有 $a \notin K$：

$$a \in J \Leftrightarrow f(a) \in \sharp T \Rightarrow f(a) \in \sharp\mathrm{Th}\,\mathfrak{N} \Leftrightarrow a \in \overline{K}.$$

因此 J 是 $\overline{K}$ 的递归可枚举子集，并且是真子集，因为 $\overline{K}$ 本身不是递归可枚举的. 也就是说，存在某个数 q 使得 $q \in \overline{K}$ 并且 $q \notin J$. 进而，$f(q) \notin \sharp\mathrm{Th}\mathfrak{N}$ 但 $f(q) \notin \sharp T$. 即，句子 $(\neg \varkappa(\mathbf{S}^q\mathbf{0}))$(在 $\mathfrak{N}$ 中) 是真的但 T 中是假的，从而证明了 T 的不完全性. 257

这个句子"说"的是什么? 对于 q，我们可以取任意数满足 $W_q = J$. 则 $q \in \overline{K}$ 并且 $q \notin J$. 这里我们分析一下情况：

$(\neg \varkappa(\mathbf{S}^q\mathbf{0}))$	表示 $q \notin K$
	即 $q \notin W_q$
	即 $q \notin J$ 因为 $W_q = J$
	即 $f(q) \notin \sharp T$ 根据 J 的定义
	即 $T \nvdash (\neg \varkappa(\mathbf{S}^q\mathbf{0}))$

用来证明 T 的不完全性的句子竟然断言它自已在公理化理论 T 中是不可证明的!

用可计算性方法和自代入法来证明哥德尔不完全性定理与上述方法没有太多不同. 不过，可计算性方法与 (3.0 节的) 对角线法更接近些.

3.6.3 判定问题的归约

假设我们已知一个二元部分递归函数 f. 例如，我们可以断定由 $g(a) = f(3,a)$ 定义的函数 g 也是部分递归函数. 根据可计算性这一非正规概念，这个结果是显然的. 要计算 g 我们只要将 f 的第一个变元固定为 3，然后根据 f 的指令进行就行了. 形式化的证明只需将这个过程表示出来就行了. 我们知道存在某个公式 $\varphi = \varphi(v_1, v_2, v_3)$ 在 $\mathrm{Cn}A_E$ 中弱表示 f(作为一个关系). 如果用 v_1, v_2 代换 φ 中的 v_2, v_3(不然，我们可以用 φ 中不出现的字母来代换)，那么 g 由公式 $\varphi(\mathbf{S}^3\mathbf{0}, v_1, v_2)$ 弱表示.

所有这些并不难懂. 当回顾已经学过的内容时，我们能够认识到一些更细致的东西. 现在我们能有效地把 f 的指令转换成 g 的指令，因此应该存在一个递归函数，使得对于给定的 f 的指标和数字 3，能够得出递归函数 g 的指标. 下面的定理是这个事实的形式表述，有时它被称为"S-m-n 定理".

参数定理 对于每个 $m \geqslant 1$，$n \geqslant 1$，存在一个递归函数 ρ 使得对于任意的 e，$\vec{a}$ 和 $\vec{b}$，有

$$[\![e]\!]_{m+n}(a_1, \cdots, a_m, b_1, \cdots, b_n) = [\![\rho(e, a_1, \cdots, a_m)]\!]_n(b_1, \cdots, b_n).$$

(此处的等号意味着，如果一边有定义，那么另一边同样也有，并且值相等. 有时我们用"$\simeq$"来表示这种用法.) 258

在等式 $\vec{a}$ 的左边是函数 $[\![e]\!]_{m+n}$ 的组成部分；而 $\vec{a}$ 的右边，是函数 $[\![\rho(e,\vec{a})]\!]_n$ 所依赖的参数. 在上面的例子中，$m = n = 1$ 并且 $a_1 = 3$. 由于 ρ 依赖于 m 和 n 两个参数，逻辑上，我们用符号"ρ_n^m"来表示. 但实际中，我们只简单地使用"ρ".

证明 $m = n = 1$ 的情况. 根据定理之前的讨论，我们可以给出这个定理的讨论. 但为了避免变量的重复，我们将采用一种略微不同的方法.

由范式定理可知，由

$$h(e, a, b) = [\![e]\!]_2(a, b)$$

定义的三元部分函数 h 是部分递归函数. 因此, 存在一个公式 ψ 弱表示 h(作为一个关系). 假设 ψ 中的 v_1, v_2 是无量词, 那么我们取

$$\begin{aligned}\rho(e,a) &= \sharp\psi(\mathbf{S}^e\mathbf{0},\mathbf{S}^a\mathbf{0},v_1,v_2)\\ &= \mathrm{Sb}(\mathrm{Sb}(\mathrm{Sb}(\mathrm{Sb}(\sharp\psi,\sharp v_1,\sharp\mathbf{S}^e\mathbf{0}),\sharp v_2,\sharp\mathbf{S}^e\mathbf{0}),\sharp v_3,\sharp v_1),\sharp v_4,\sharp v_2).\end{aligned}$$

则 $\rho(e,a)$ 是一个公式的哥德尔数, 这个公式弱表示函数 $g(b)=[\![e]\!]_2(a,b)$. 因此, 它是 g 的指标.

■

参数定理可以用来证明某些集合不是递归的. 我们已经知道集合 $K=\{a|[\![a]\!]_1(a)$ 是可定义的 $\}$ 不是递归的. 对于一个给定的非递归集 A, 有些时候我们能够找到一个 (全) 递归函数 g 使得

$$a\in K\Leftrightarrow g(a)\in A$$

或者 (全) 递归函数 g' 使得

$$a\notin K\Leftrightarrow g'(a)\in A.$$

无论哪种情况, 我们都能得到 A 不是递归的. 否则的话, K 将是递归的. 在前一种情况中, 我们有 $K\leqslant_m A$ 并且 A 不属于 Π_1(根据引理 36I); 在后一种情况中, $\overline{K}\leqslant_m A$ 并且 A 不属于 Σ_1. 在每种情况中, A 都不是递归的. 函数 g 或 g' 通常可以由参数定理得到.

例 $\{a|W_a=\varnothing\}$ 不是递归的.

证明 称这个集合为 A. 首先注意到 $A\in\Pi_1$, 这是因为 $W_a=\varnothing$ 当且仅当 $\forall b\forall k\langle a,b,k\rangle\notin T_1$. 进而, K 不能多一归约于 A, 但我们有理由认为 $\overline{K}$ 是多一归约到 A 的. 即, 我们要找一个全递归函数 g 使得

259

$$[\![a]\!]_1(a)\text{是不可定义的}\Leftrightarrow \mathrm{dom}[\![g(a)]\!]_1=\varnothing$$

如果对于所有的 b, $[\![g(a)]\!]_1(b)=[\![a]\!]_1(a)$ 成立, 那么这个等价式成立. 所以我们从部分递归函数 $f(a,b)=[\![a]\!]_1(a)$ 开始, 令 $g(a)=\rho(\hat{f},a)$, 其中 $\hat{f}$ 是 f 的指标. 那么

$$[\![g(a)]\!]_1(b)=[\![\rho(\hat{f},a)]\!]_1(b)=f(a,b)=[\![a]\!]_1(a).$$

这样, g 就证明了 $\overline{K}$ 是多一归约于 A 的. ■

定理 36J(Rice, 1953) 设 $\mathcal{C}$ 是一元部分递归函数的集合. 那么 $\mathcal{C}$ 中元素的指标集 $\{e|[\![e]\!]_1\in\mathcal{C}\}$ 是递归的当且仅当要么 $\mathcal{C}$ 是空的要么 $\mathcal{C}$ 包含所有一元部分递归函数.

证明 我们只需要证明一个方向. 令 $I_\mathcal{C}=\{e|[\![e]\!]_1\in\mathcal{C}\}$ 为 $\mathcal{C}$ 中元素的指标集.

情形 I: 空函数 $\varnothing$ 不在 $\mathcal{C}$ 中. 如果 $\mathcal{C}$ 中什么也没有, 那么证明完毕. 现在假设函数 ψ 在 $\mathcal{C}$ 中, 我们能够证明, 如果递归全函数 g 满足

$$[\![g(a)]\!]_1=\begin{cases}\psi & \text{如果}a\in K,\\ \varnothing & \text{如果}a\notin K.\end{cases}$$

那么 K 多一归约于 $I_\mathcal{C}$. 这是因为 $a\in K\Leftrightarrow[\![g(a)]\!]_1\in\mathcal{C}\Leftrightarrow g(a)\in I_\mathcal{C}$.

我们可以利用参数定理定义

$$g(a) = \rho(e, a)$$

其中

$$[\![e]\!]_2(a,b) = \begin{cases} \psi(b) & \text{如果} a \in K, \\ \text{无定义} & \text{如果} a \notin K. \end{cases}$$

由于 $[\![e]\!]_2(a,b) = c \Leftrightarrow a \in K$ 且 $\psi(b) = c$, 并且等价条件的右侧是递归可枚举的, 因此上式是一个部分递归函数.

情形 II: $\varnothing \in \mathcal{C}$. 我们把情形 I 的证明运用到 $\mathcal{C}$ 的补集 $\overline{\mathcal{C}}$ 上. 我们能够断定 $I_{\overline{\mathcal{C}}}$ 不是递归的. 但 $I_{\overline{\mathcal{C}}}$ 是 $I_{\mathcal{C}}$ 的补集, 因此 $I_{\mathcal{C}}$ 不可能是递归的.

这样, 在两种情况中, $I_{\mathcal{C}}$ 都不是递归的. ■

例 对于任意取定的 e, 根据 Rice 定理的结论, 集合 $\{a|W_a = W_e\}$ 不是递归的. 特别地, $\{a|W_a = \varnothing\}$ 不是递归的, 这是前一个例子的结论. 对于 Rice 定理的另外两个应用, 集合 $\{a|W_a\text{是无限的}\}$ 和 $\{a|W_a\text{是递归的}\}$ 都不是递归的.

260

3.6.4 带寄存的计算器

递归函数类有许多等价的定义, 其中几个定义被用在理想化的计算设备中. 这些计算设备, 如数字计算机, 没有内存空间的限制. 这类定义中的第一个是由 Alan Turing 在 1936 年提出的; 几乎在同一时间, Emil Post 也做了类似的工作. 我们在这里将要介绍的定理是 Shepherd- son 和 Sturgis 在 1963 年提出的, 与前两位的结论有所不同.

带寄存的计算器 只有有限多个计数器, 分别标以 $1, 2, \cdots, K$. 每个计数器能储存一个任意大的自然数. 带寄存的计算器的运算由程序来决定. 一个程序由有限的 *指令* 序列组成.

Ir($1 \leqslant r \leqslant K$), 表示 "$r$ 增加. " 这个指令是让计数器 r 中的数加 1. 然后带寄存的计算器进入程序的下一条指令.

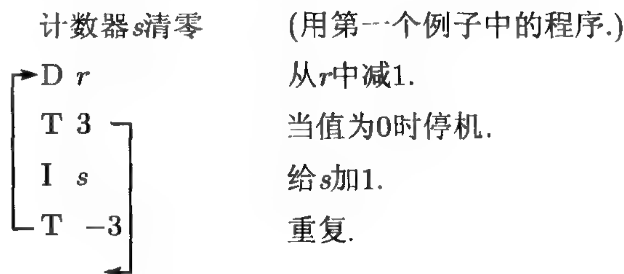
Dr($1 \leqslant r \leqslant K$), 表示 "$r$ 减少. " 这个指令依赖于计数器 r 中的数. 如果 r 中的数不是零, 那么将其减 1, 然后跳过下一条指令进入随后的一条指令. 如果 r 中的数是零, 那么带寄存的计算器将执行下一条指令. 概括地说: 带寄存的计算器试图对计数器 r 中的数减 1, 如果执行了这一步, 那么跳过下一步的指令.

Tq(q 是一个整数 - 正数, 负数或零), 表示 "转到 q. " 所有的计数器保持不变. 带寄存的计算器执行程序中这条指令后的第 q 条指令 (如果 $q \geqslant 0$), 或者这条指令前的第 $|q|$ 条指令 (如果 $q < 0$). 如果程序中不存在符合条件的指令, 带寄存的计算器就停机. 指令 T0 将得到一个循环, 即带寄存的计算器反复地执行这个指令.

例 (1) 计数器 7 清零的程序.

```
┌→D 7      对计数器7中的数减1.
│ T 2 ┐
└─T −2│    回到第一步并重复.
     ←┘    停机.
```

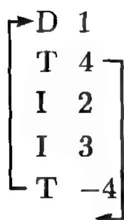
(2) 将一个数从计数器 r 移至计数器 s 的程序.



261

这个程序总共有 7 条指令. 计数器 r 中的数最后为 0.

(3) 把计数器 1 中的数加到计数器 2 和计数器 3 中.



(4) (加法) 假设计数器 1 和 2 中的数分别为 a 和 b. 我们要把 $a+b$ 存入计数器 3, 并且要求最后计数器 1 和 2 中的数仍为 a 和 b.

	计数器的值			
计数器 3 清零	a	b	0	
把计数器 1 中的数移至计数器 4	0	b	0	a
把计数 4 中的数加到计数器 1 和 3	a	b	a	0
把计数器 2 中的数移至计数器 4	a	0	a	b
把计数 4 中的数加到计数器 2 和 3	a	b	$a+b$	0

如果把这个程序完整地写出来, 它有 27 个指令, 但其中 3 个是多余的. (我们在第 4 行把计数器 4 清零.) 最后我们将 a 加回计数器 1. 但在这个过程中, 计数器 1 必须先清零, 这是确保计数器 1 中的数是 a 的唯一办法.

(5) (减法) 令 $a \dot{-} b = \max(a-b, 0)$. 我们把这个程序留给读者来完成 (习题 11).

现在假设 f 是 $\mathbb{N}$ 上的 n 元部分函数. 可能存在一个程序 P 使得, 如果我们把 $a_1, \cdots, a_n$ 存入一个带寄存的计算器的计数器 $1, \cdots, n$ 中 (假设带寄存的计算器拥有 P 所需要的所有计数器), 并运行程序 P, 那么下列的条件成立:

(i) 如果 $f(a_1, \cdots, a_n)$ 是可定义的, 则计算将最终在计数器 $n+1$ 上写上 $f(a_1, \cdots, a_n)$, 而且计算在执行第 $(p+1)$ 步指令时停止, 这儿 p 是 P 的长度.

(ii) 如果 $f(a_1, \cdots, a_n)$ 是不可定义的, 那么计算不会停止.

如果这样的程序 P 存在, 我们称 P计算f.

定理 36K　设 f 是部分函数, 则存在一个程序计算 f 当且仅当 f 是部分递归函数.

这样, 利用带寄存的计算器我们确切得到了部分递归函数类, 这个函数类起先是根据有

262

限公理化的和谐理论中的可表示性来定义的. 我们采用不同的方法都得到了部分递归函数类

这一相同的对象，这正说明了这个函数类的重要性.

证明 为了证明带寄存的计算器可计算的函数是部分递归函数，我们要用 3.4 节中算术化推理的方法来算术化计算. 即，把哥德尔数分配给程序以及存储器序列. 我们要证明，所有相关的概念通过哥德尔数转换成数字关系后都是递归的. (完成这些之后，我们认识到，从广义的角度说，推理和计算本质上是相同的.)

反之，为了证明部分递归函数可以通过带寄存的计算器来计算，我们要重复 3.3 节和 3.4 节的内容，只不过当时要证明的是函数在 CnA_E 中可表示，现在我们要证明它们可以由带寄存的计算器计算. 这并没有想象得那么难，证明的过程和前面是一样的. 这是因为我们可以证明，递归函数类是由一些递归函数通过复合运算 (定理 33L) 和“最小零”运算 (定理 33M) 生成的. 这个结果实际上可以从 3.3 节和 3.4 节中的工作中得到. 这样，一旦我们证明了构成递归函数类的初始函数可以通过带寄存的计算器来计算，并且这些函数在复合运算和“最小零”运算下是封闭的，那么重复前面类似的证明，我们就能得到所有递归函数的可计算性. ■

习题

1. 如下定义函数 f 和 g,

$$f(n) = \begin{cases} 0 & \text{如果哥德巴赫猜想是对的,} \\ 1 & \text{其他情况.} \end{cases}$$

$$g(n) = \begin{cases} 0 & \text{如果在 } \pi \text{ 的十进制小数展开式中至少连续出现 } n \text{ 个 } 7, \\ 1 & \text{其他情况.} \end{cases}$$

那么，f 是递归的吗？g 是递归的吗？(哥德巴赫猜想是指任意一个比 2 大的偶数都可以写成两个素数的和. 在这本书的第一版使用的是 Fermat 最终定理.)

263

2. 定义“对角线”函数 $d(a) = [\![a]\!]_1(a) + 1$.
   (a) 证明：d 是部分递归函数.
   (b) 根据 (a)，我们知道存在某个数 e，使得 $d = 1$. 因此，一方面 $d(e) = [\![a]\!]_1(e)$，另一方面 $d(e) = [\![a]\!]_1(e)+1$. 我们是否能通过移项得到 0=1？提示：用“$\simeq$”来表示要么方程的两边都没定义，要么两边都有定义并且相等. 用这个符号重新表示这个问题.
3. (a) 证明：任意部分递归函数的值域都是递归可枚举的.
   (b) 证明：严格递增的全递归函数 f(即，$f(n) < f(n+1)$) 的值域是递归的.
   (c) 证明：递增全递归函数 f(即，$f(n) \leqslant f(n+1)$) 的值域是递归的.
4. (a) 设 A 是 $\mathbb{N}$ 的非空递归可枚举子集. 证明 A 是某个全递归函数的值域.
   (b) 证明：$\mathbb{N}$ 的任意无限递归可枚举子集都包含一个无限递归子集.
5. 证明：每个部分递归函数都有无穷多个指标.
6. 找出一个函数 f 和数 e，使得对于所有 a,

$$f(a) = U(\mu k\langle e, a, k\rangle \in T_1)$$

7. 证明：参数定理的条件中的 ρ 可以加强为一对一的.
8. 已知两个递归可枚举的并是递归可枚举的 (3.5 节的习题 7). 证明：存在一个全递归函数 g 使得 $W_{g(a,b)} = W_a \cap W_b$.

9. 证明：集合 $\{a|W_a$ 包含两个 (以上) 元素 $\}$ 在 Σ_1 中但不在 Π_1 中.

10. 证明：不存在递归可枚举集 A 使得 $\{[\![a]\!]_1|a\in A\}$ 等于 $\mathbb{N}$ 上的全递归函数类.

11. 给出计算下列函数的带寄存的计算器程序：

 (a) 减法，$a\dot{-}b=\max(a-b,0)$.

 (b) 乘法，$a\cdot b$.

 (c) $\max(a,b)$.

12. 假设存在一个计算 n 元部分函数 f 的带寄存的计算器程序. 证明，对于任意给定的正整数 $r_1,\cdots,r_n$(各不相同)，p 和 k，我们能够找到程序 Q，使得只要我们把 $a_1,\cdots,a_n$ 存入一个带寄存的计算器的计数器 $r_1,\cdots,r_n$ 中 (假设带寄存的计算器拥有 Q 所需要的所有计数器)，并运行程序 Q，那么

 (i) 如果 $f(a_1,\cdots,a_n)$ 有定义，则计算将在计数器 p 的 $f(a_1,\cdots,a_n)$ 处停止，计数器 $1,2,\cdots,k$(除了计数器 p) 中的数与开始时相同，并且计算在执行第 $(q+1)$ 步指令时停止，其中 q 是 Q 的长度；

 (ii) 如果 $f(a_1,\cdots,a_n)$ 没有定义，则计算永远不会停止.

264

13. 令 $g:\mathbb{N}^{n+1}\longrightarrow\mathbb{N}$ 是一个 (全) 函数，它由某个带寄存的计算器程序计算. 设 $f(a_1,\cdots,a_n)=\mu b[g(a_1,\cdots,a_n,b)=0]$，其中如果没有满足条件的 b 存在，那么等号右边无意义. 证明部分函数 f 能够由某个带寄存的计算器程序计算.

14. 证明下列集合所属的算术分层. (在每种情况中，给定的位置可能是最佳的，但我们并不要求证明.)

 (a) $\{e|[\![e]\!]_1$ 是全函数 $\}$ 是 Π_2.

 (b) $\{e|W_e$ 是有限的 $\}$ 是 Σ_2.

 (c) $\{e|W_e$ 是余有限的 $\}$ 是 Σ_3.

 (d) $\{e|W_e$ 是递归的 $\}$ 是 Σ_3.

15. 设 $Tot=\{e|[\![e]\!]_1$ 是全函数 $\}$，显然 $Tot\subset K$. 证明不存在递归集 A 满足

$$Tot\subseteq A\subseteq K.$$

 说明：这个结果包含了定理 36E 和 36F，这两个定理的证明方法同样可以用在这里.

16. (a) 证明：对于每个自然数集 Π_2，存在某个数 e，使得这个集合等于集合

$$\{a|\forall b\exists cT_2(e,a,b,c)\}.$$

 (b) 证明：集合 $\{a|$ 非 $\forall b\exists cT_2(e,a,b,c)\}$ 是 Σ_2 但不是 Π_2.

 *(c) 推广 (a) 和 (b) 来证明对于每个 n，存在一个集合是 Σ_n 但不是 Π_n.

17. 假设 A 是自然数的集合，它是算术的但不是 Π_m. 利用 192 页的讨论来证明 $\sharp\mathrm{Th}\mathfrak{N}$ 不是 Σ_m. 说明：习题 16 和 17 给出了从可计算性理论来证明塔斯基定理 (即 $\sharp\mathrm{Th}\mathfrak{N}$ 不是算术的) 的一种方法.

265

3.7 第二不完全性定理

现在让我们回到 3.4 节中的第 20 条. 假设递归可公理化理论 T 由递归公理集 A 给出 (即，$\sharp A$ 是递归的). 那么正如 20 条所述：

$a\in\sharp T\Leftrightarrow d[d$ 是从 A 出发的一个推理的哥德尔数，
并且 d 的最后一个成分 a 是一个句子的哥德尔数].

序对 $\langle a,d\rangle$ 满足方括号中的条件，这些序对的集合是递归的. 令 $\pi(v_1,v_2)$ 是在 A_E 中数字表示这个二元关系的公式.

对于任意句子 σ，我们可以用 $\exists v_2\pi(\mathbf{S}^{\sharp\sigma}\mathbf{0},v_2)$ 表示 "$T\vdash\sigma$". 为了叙述方便，定义

$$\mathrm{Prb}_T\sigma = \exists v_2 \pi(\mathbf{S}^{\sharp\sigma}\mathbf{0}, v_2).$$

(此处 Prb 是 "provable" 的缩写. 在构造上面的句子时, 我们用到了公理集 A 的递归性, 因此, 写在下方的似乎应该是 "A" 而不是 "T".)

引理 37A 设 T 是如上所述的递归可公理化理论.

(a) 只要 $T \vdash \sigma$ 则 $A_E \vdash \mathrm{Prb}_T\sigma$.

(b) 如果 T 包含 A_E, 那么 T 有 "反射" 性质:

$$T \vdash \sigma \Rightarrow T \vdash \mathrm{Prb}_T\sigma.$$

证明 如果 $T \vdash \sigma$, 那么我们可以令 d 是从 T 的公理 A 到 σ 的推理所对应的哥德尔数. 这样就有 $A_E \vdash \pi(\mathbf{S}^{\sharp\sigma}\mathbf{0}, \mathbf{S}^d\mathbf{0})$, 进而 $A_E \vdash \mathrm{Prb}_T\sigma$, 这就证明了 (a), 从 (a) 我们可以立即得到 (b). ■

这样, 在适当的假设下, 只要 T 证明了一个句子, 那么 T 就知道它证明了这个句子. 请注意, (b) 并不意味着 $T \vdash (\sigma \to \mathrm{Prb}_T\sigma)$. 例如, σ(在 $\mathfrak{N}$ 中) 是真的但不能从 A_E 中得到证明, 那么句子 $(\sigma \to \mathrm{Prb}_T\sigma)$ 就不能从 A_E 中得到证明, 实际上它在 $\mathfrak{N}$ 中是假的.

现在回到哥德尔不完全性定理的证明 (在自代入法中) 中来, 我们可以利用不动点引理得到句子 σ, 它断言自己在 T 中不能被证明:

$$A_E \vdash (\sigma \leftrightarrow \neg\mathrm{Prb}_T\sigma).$$

下面的引理是不完全性定理的一部分内容 (另一部分内容是 3.5 节的习题 2). 266

引理 37B 设 T 是包含 A_E 的递归可公理化理论, σ 是利用不动点引理得到的上述句子. 如果 T 是和谐的, 则 $T \nvdash \sigma$.

证明

$$\begin{aligned} T \vdash \sigma &\Rightarrow T \vdash \mathrm{Prb}_T\sigma && \text{根据反射性} \\ &\Rightarrow T \vdash \neg\sigma && \text{根据 } \sigma \text{ 的选取} \end{aligned}$$

这样 T 是不和谐的. ■

到目前为止, 这个引理只是反映了 3.5 节中的思想, 并且它的证明并不复杂. 也正是由于这一点, 如果 T "足够强", 证明有可能在理论 T 中实现. 即, 我们希望步骤

$$\begin{aligned} \mathrm{Prb}_T\sigma &\to \mathrm{Prb}_T\mathrm{Prb}_T\sigma \\ &\to \mathrm{Prb}_T\neg\sigma \\ &\to \mathrm{Prb}_T\mathbf{0} = \mathbf{S0} \end{aligned}$$

能够在 T(包含 A_E) 的一个足够强的扩充中实现.

如果这样, 我们就能得到一个不同寻常的结论. 为此, 令 $\mathrm{Cons}T$ 是句子 $\neg\mathrm{Prb}_T\mathbf{0} = \mathbf{S0}$, 它表示 "$T$ 是和谐的. " (这里取 $\mathbf{0} = \mathbf{S0}$ 只是为了选一个被 A_E 反驳的简单公式.) 如果上一段中的步骤能够在 T 中实现, 那么我们有:

$$T \nvdash \mathrm{Cons}T, \text{除非 } T \text{ 是不和谐的}.$$

(当然，一个不和谐的理论可以包含每个句子，包括错误地断定这个理论是和谐的句子. 我们在此希望能够找到，在合适的假设下，理论证明自己的和谐性的唯一方法.) 让我们再回顾一下细节：假设 $T \vdash \mathrm{Cons}T$，那么根据前一段的推导有 $T \vdash \mathrm{Prb}_T\sigma$. 由 σ 的选取，我们可得 $T \vdash \sigma$. 接下去我们可以使用引理 37B.

为了使表达更加清楚，我们把满足下列 3 个“可推导性”条件的理论 T 称为**足够强**.

(1) $A_E \subseteq T$. 根据引理 37A, T 有反射性质, $T \vdash \sigma \Rightarrow T \vdash \mathrm{Prb}_T\sigma$.

(2) 对于任意句子 σ, $T \vdash (\mathrm{Prb}_T\sigma \to \mathrm{Prb}_T\mathrm{Prb}T\sigma)$. 这是 T 中反射性质的形式化.

(3) 对于任意句子 ρ 和 σ, $T \vdash (\mathrm{Prb}_T(\rho \to \sigma) \to (\mathrm{Prb}_T\rho \to \mathrm{Prb}_T\sigma))$. 这是 T 中假言推理的形式化.

引理 37B 的形式化　假设 T 是足够强的递归可公理化理论, σ 是满足下列式子的句子：

$$A_E \vdash (\sigma \leftrightarrow \neg\mathrm{Prb}_T\sigma).$$

267　则 $T \vdash (\mathrm{Cons}T \to \neg\mathrm{Prb}_T\sigma)$.

证明　我们把上面叙述的内容归纳起来. 首先由 σ 的选取得

$$T \vdash (\sigma \to (\mathrm{Prb}_T\sigma \to \mathbf{0} = \mathbf{S0})).$$

然后对这个公式运用第一个反射性质和形式化的假言推理，得到

$$T \vdash (\mathrm{Prb}_T\sigma \to \mathrm{Prb}_T(\mathrm{Prb}_T\sigma \to \mathbf{0} = \mathbf{S0}))$$

再使用一次形式化的假言推理，我们有

$$T \vdash (\mathrm{Prb}_T\sigma \to (\mathrm{Prb}_T\mathrm{Prb}_T\sigma \to \neg\mathrm{Cons}T)).$$

这个公式 ($\vdash$ 的右边部分) 与 $\mathrm{Prb}_T\sigma \to \mathrm{Prb}_T\mathrm{Prb}_T\sigma$(形式化的反射性质) 一起，通过命题逻辑推出 $\mathrm{Prb}_T\sigma \to \mathrm{Cons}T$. ■

哥德尔第二不完全性定理 (1931)　设 T 是一个足够强的递归可公理化理论，则 $T \vdash \mathrm{Cons}T$ 当且仅当 T 是不和谐的.

证明　如果 $T \vdash \mathrm{Cons}T$，并且由 σ 的选择，我们知 $T \vdash \sigma$. 根据引理 37B 的形式化，我们有 $T \vdash \mathrm{Prb}_T\sigma$. 从 (未形式化的) 引理 37B 可得 T 是不和谐的. ■

我们可以从这些思路中得到一点启发. 实际上引理 37B 可以看作下面这个引理 (当 τ 是 $\mathbf{0}{=}\mathbf{S0}$) 的一个特例.

引理 37C　设 T 是包含 A_E 的递归可公理化理论, τ 是句子, σ 是通过不动点引理得到的句子，满足下列公式：

$$A_E \vdash (\sigma \leftrightarrow (\mathrm{Prb}_T\sigma \to \tau)).$$

如果 $T \vdash \sigma$，那么 $T \vdash \sigma$.

证明　我们可以认为 σ 叙述的是“如果我是可证明的，那么 τ. ”如果 $T \vdash \sigma$，根据反射性质，有 $T \vdash \mathrm{Prb}_T\sigma$. 再由 σ 的选择，我们有 $T \vdash \tau$. ■

实际上我们感兴趣的不是这个引理，而是它的形式化：

引理 37C 的形式化 设 T 是足够强的递归可公理化理论，τ 是句子，σ 是通过不动点引理得到的句子，满足下列公式：

$$A_E \vdash (\sigma \leftrightarrow (\mathrm{Prb}_T\sigma \to \tau)).$$

则 $T \vdash (\mathrm{Prb}_T\sigma \to \mathrm{Prb}_T\tau)$.

268

证明 过程和前面一样. 由 σ 的选择，我们有

$$T \vdash (\sigma \to (\mathrm{Prb}_T\sigma \to \tau)).$$

对这个公式运用第一个反射性质和形式化的假言推理，得到

$$T \vdash (\mathrm{Prb}_T\sigma \to \mathrm{Prb}_T(\mathrm{Prb}_T\sigma \to \tau))$$

再使用一次形式化的假言推理，我们有

$$T \vdash (\mathrm{Prb}_T\sigma \to (\mathrm{Prb}_T\mathrm{Prb}_T\sigma \to \mathrm{Prb}_T\tau)).$$

这个公式 ($\vdash$ 的右边部分) 与 $\mathrm{Prb}_T\sigma \to \mathrm{Prb}_T\mathrm{Prb}_T\sigma$(形式化的反射性质) 一起，通过命题逻辑推出 $\mathrm{Prb}_T\sigma \to \mathrm{Prb}_T\tau$. ■

Löb 定理 (1955) 设 T 是足够强的递归可公理化理论，如果 τ 是满足 $T \vdash (\mathrm{Prb}_T\tau \to \tau)$ 的任意句子，那么 $T \vdash \tau$.

显然，如果 $T \vdash \tau$，那么对于任意句子 ρ 有 $T \vdash (\rho \to \tau)$. 所以 Löb 定理的结论可以表述为 $T \vdash (\mathrm{Prb}_T\tau \to \tau) \iff T \vdash \tau$.

证明 对于给定的句子 τ，构造 σ 为“如果我是可证明的，那么 τ. ”假设 $T \vdash (\mathrm{Prb}_T\tau \to \tau)$，根据形式化的引理 37C，我们有 $T \vdash (\mathrm{Prb}_T\sigma \to \mathrm{Prb}_T\tau)$. 由 σ 的选择，有 $T \vdash \sigma$. 再根据 (未形式化的) 引理 37C，我们有 $T \vdash \tau$. ■

Löb 定理开始是为了解决习题 1 的问题而提出的. 但它导出了 (从某种意义上说是等价的) 哥德尔第二不完全性定理. 假设 T 是足够强的递归可公理化理论，运用 Löb 定理并将 τ 取为 $\mathbf{0} = \mathbf{S0}$, 那么

$$T \vdash (\mathrm{Prb}_T(\mathbf{0} = \mathbf{S0}) \to \mathbf{0} = \mathbf{S0}) \Rightarrow T \vdash \mathbf{0} = \mathbf{S0},$$

即，

$$T \vdash \mathrm{Cons}T \Rightarrow T\text{是不和谐的}.$$

这样，我们就得到了第二不完全性定理.

现在还有一个问题没有回答：什么样的理论是足够强的？这样的理论是否存在 (除了不和谐理论这个平凡的情况)？

答案是肯定的并且我们有两个理论. 第一个是“皮亚诺代数” (PA). 它的公理集包括 A_E 公理和所有的“归纳公理”. 对于一个合式公式 φ，存在万有闭包，闭包中的公式都具有下列形式：

269

$$\varphi(0) \land \forall x(\varphi(x) \to \varphi(\mathbf{S}x)) \to \forall x\varphi(x)$$

归纳公理描述了数学归纳法的一般原则，它使我们能够在皮亚诺代数中讨论自然数的许多问题 (例如，加法的交换律). 但要确保形式化的反射性和假言推理也可以从皮亚诺代数得到，需要证明许多细节，在此我们就不一一证明了.

我们知道皮亚诺代数是和谐的，因为它在 $\mathfrak{N}$ 中是真的. 但根据第二不完全性定理，PA 不能证明自身的和谐性. 通过在非形式化数学或集合论中的讨论，我们“知道”PA 是和谐的. 因此集合论比 PA 有更高的“和谐性能力”：它证明了 PA 的和谐性但 PA 不能.

第二个足够强的理论是公理集合论. 更细致地说，它是能够在公理集合论中被证明的数论语言中的句子集. 下一小节就将讨论这个问题. 从非形式化的角度看，我们从这个理论可以诱导出形式化的反射性和假言推理. 但集合论自身的和谐性又从哪里来? 我们知道 PA 的和谐性是由于它在数论的“标准模型”中是真的. 但我们根本说不清“集合论的标准模型”!

3.7.1　集合论的应用

我们已经知道，在数论语言中，$\mathrm{Cn}A_E$ 不是完全的也不是递归的，其他在这个语言中递归可公理化的和谐理论也是一样.

现在把目光从算术转到集合论. 我们已知一种语言 (包含参数 $\forall$ 和 $\in$) 和一个公理集. 在目前所有公认的例子中，集合论的公理集是递归的. 更准确地说，应该是公理的哥德尔数集是递归的. 因此得到的理论 (集合论) 是递归可枚举的. 我们断言，这个理论如果是和谐的，它就不是递归的，进而不是完全的. 我们先大体地了解一下讨论过程. 首先，数论的语言可以被嵌入集合论，但我们只关心与自然数及其运算有关的那部分 (图 3-5 中的阴影部分)，即与 A_E 和谐的理论. 它不是递归的，因为如果集合论是递归的，则它的算术部分也是递归的，但事实上不是这样. 了解了这些后，我们将进入第二不完全性定理在集合论中的情况.

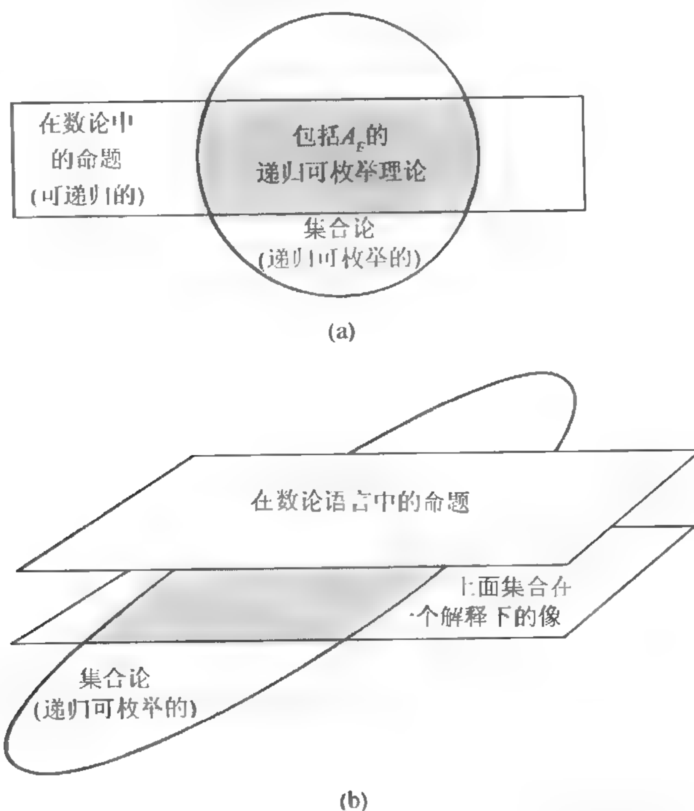


图 3-5　集合论和数论：(a) 平面图，(b) 更精确的图

今后，我们所说的集合论 (ST) 是指大家都熟悉的 (在含有等于及 $\forall$ 和 $\in$ 两个参数的语言中) 集合理论的公理序列集 (标准的策梅洛 - 弗兰克尔公理也行，如果大家熟悉．我们只要求公理集是递归的，并且从它能够得到我们现在已知的关于集合的事实)．我们需要从 $\mathrm{Cn}A_E$ 到 ST 的解释．(这一节的剩余部分与 2.7 节有相同的假设．) 但这个 π 的存在要依赖于集合论的结果，而不是这几句话能够说明的．我们需要用 ST 语言中的公式来表示自然数的概念，两个数的和的概念，等等．为了找到这些公式，我们重新考虑自然数运算 "嵌入" 集合论的方法．也就是，一方面，2, 7 这样的自然数不再以集合的身份出现；另一方面，当我们需要时可以选取适当的集合来表示数．标准的方法是把 0 对应到集合 $\varnothing$，$n+1$ 对应到集合 $n;n$．这么做的另一个好处是每一个数都是所有比它小的数组成的集合 (如 $3\in 7$)．令 ω 是所有这些数集的并，那么 ω 是表示 $\mathbb{N}$ 的集合．

[270]

[271]

从公式 $v_1 \in \omega$ 中去掉已定义的 ω 后得到的公式用 $\pi_\forall$ 表示．类似地，π_0 是由公式 $v_1=\varnothing$ 得到的，公式 $\pi_\mathbf{S}$ 由 $v_2 = v_1 \cup \{v_1\}$ 得到．我们用公式 $\pi_<$ 简单地表示 $v_1 \in v_2$，用下面的句子在 ST 语言中的翻译来表示 π_+：

对于任意 f，如果 $f:\omega\times\omega\to\omega$，并且对于所有 ω 中的 a 和 b，有 $f(a,\varnothing)=a$，并且 $f(a,b\cup\{b\})=f(a,b)\cup\{f(a,b)\}$，那么 $f(v_1,v_2)=v_3$.

(第 0 章中介绍了部分翻译的规则．) 用同样的方式，我们可以得到公式 $\pi_.$ 和 $\pi_\mathbf{E}$.

那么，π 是从 $\mathrm{Cn}A_E$ 到 ST 的解释，这个断言在 ST 上成立必须满足下列的数字 (数字是 17) 要求．

(i) $\exists v_1 \pi_\forall$ 必定在 ST 中，因为在集合论中，ω 是非空的．

(ii) 用 f 表示语言 A_E 的 5 个函数符号之一，ST 必然包含一个句子，这个句子断言 π_f 在 $\pi_\forall$ 定义的集合上定义了一个函数．(这个句子就是 2.7 节中 "解释" 的定义中提出的句子．) 在 $\mathbf{0}$ 的情况下，结果是在 ST 中存在唯一一个空集，这个空集属于 ω．$\mathbf{S}$ 的情况也很简单，由于 $\pi_\mathbf{S}$ 在所有集合的论域中定义了一个一元运算，并且 ω 在这个运算下是封闭的．对于 $+$，我们必须对 ω 使用递归定理．即，在 ST 中证明 (如 1.4 节中提到) 存在唯一的 $f:\omega\times\omega\to\omega$ 使得对于所有 ω 中的 a 和 b，有 $f(a,\varnothing)=a$ 并且 $f(a,b\cup\{b\})=f(a,b)\cup\{f(a,b)\}$．这样，$\pi_+$ 所要求的性质就得到了．对 $\bullet$ 和 $\mathbf{E}$ 可以进行类似的讨论．

(iii) 用 σ 表示 A_E 中的 11 个句子之一，那么 σ^π 必然在 ST 中．例如 L3 的情况，对于 ω 中任意的 m 和 n，我们有 $m\in n$，$m=n$ 和 $n\in m$ 必有一个在 ST 中．

由于这些要求都是有限数，因此存在有限 $\Phi\subseteq \mathrm{ST}$，使得 π 也是 $\mathrm{Cn}A_E$ 到 $\mathrm{Cn}\Phi$ 的解释．

定理 37D(集合论的强不可判定性) 设 T 是集合论语言中的一个理论满足 $T\cup\mathrm{ST}$(至少 $T\cup\Phi$) 是和谐的，那么 $\sharp T$ 不是递归的．

证明 令 Δ 为和谐理论 $\mathrm{Cn}(T\cup\Phi)$，Δ_0 为 $\pi^{-1}[\Delta]$ 在数论语言中对应的理论．由 2.7 节知，Δ_0 是一个和谐理论 (因为 Δ 是)．同时 $A_E\subseteq\Delta_0$，这是因为如果 $\sigma\in A_E$，那么$\sigma^\pi\in \mathrm{Cn}\Phi\subseteq\Delta$．因此由 $\mathrm{Cn}A_E$ 的强不可判定性 (定理 35C) 可知，$\sharp\Delta_0$ 不是递归的．

[272]

现在我们要从 Δ_0 的不可递归性得到 T 的不可递归性．我们有

$$\sigma\in\Delta_0 \quad \text{iff} \quad \sigma^\pi\in\Delta$$

并且由下面的引理可得，$\sharp\sigma^\pi$ 递归地依赖于 $\sharp\sigma$．即 $\sharp\Delta_0\leqslant_m\sharp\Delta$．因此 $\sharp\Delta$ 不可能是递归的，否

则 $\sharp\Delta_0$ 就是递归的. 类似地, 我们有

$$\tau \in \Delta \text{ iff } (\varphi \to \tau) \in T,$$

其中 φ 是 Φ 中公式的合取. 由于 $\sharp(\varphi \to \tau)$ 递归地依赖于 $\sharp\tau$, 因此我们有 $\sharp\Delta \leqslant_m \sharp T$ 使得 $\sharp T$ 不可能是递归的, 否则 $\sharp\Delta$ 就是递归的. ■

引理 37E 存在一个递归函数 p, 使得对于数论语言中的公式 α, $p(\sharp\alpha) = \sharp(\alpha^\pi)$.

证明 在 2.7 节中我们清楚地给出了构造 α^π 的过程. 在那些例子的构造过程中, 利用了比 α 简单的公式 β 所对应的 β^π. 而我们可以把 3.3 节和 3.4 节中的方法运用到这些公式的哥德尔数上, 来证明 p 是递归的. 但详细的证明并不吸引人, 在此就忽略了. ■

推论 37F 如果集合论是和谐的, 那么它不是完全的.

证明 我们知道集合论有递归的公理集. 如果集合论是完全的, 那么它就是递归的 (根据 3.4 节的 21 条). 由前面的定理知, 如果 ST 是和谐的, 这种情况不可能发生. ■

推论 37G 在含有等号和一个二元谓词符号的语言中, 永真句子 (的哥德尔数) 集合不是递归的.

部分证明 在上面的定理中取 $T = \text{Cn}\varnothing$ 为永真句子的集合. 则定理告诉我们, 如果 Φ 是和谐的, 那么 $\sharp T$ 不是递归的. 我们目前还没有明确给出有限集 Φ, 但我们可以肯定, Φ 可以通过和谐性选取出来. ■

我们必须注意, π 不是 $\text{Th}\mathfrak{N}$ 到 ST 的解释 (除非 ST 是不和谐的). 这是因为, 作为引理 37E 的结果, $\pi^{-1}[\text{ST}]$ 是语言 $\mathfrak{N}$ 中的递归可枚举理论. 因此, 它不可能等于 $\text{Th}\mathfrak{N}$, 并且只有当它是不和谐的时, 它才能包含完全理论 $\text{Th}\mathfrak{N}$.

273

3.7.2 集合论中的哥德尔第二不完全性定理

我们仍然可以用常用的方法找到一个数论的句子 σ, 它间接地表达了它本身的解释 σ^π 不是集合论中的定理. 令 D 是 $\mathbb{N}$ 上的三元关系, 满足

$$\langle a, b, c\rangle \in D \text{ iff } a \text{ 是数论中的公式}\alpha\text{的哥德尔数, 并且 } c \text{ 是从}\alpha(S^b0)^\pi\text{的 ST 的公理推理的哥德尔数.}$$

(通过常用的讨论可知) 关系 D 是递归的; 设 $\delta(v_1, v_2, v_3)$ 在 $\text{Cn}A_E$ 中表示 D, r 是公式 $\forall v_3\neg\delta(v_1, v_2, v_3)$ 的哥德尔数, σ 是公式 $\forall v_3\neg\delta(\mathbf{S}^r\mathbf{0}, \mathbf{S}^r\mathbf{0}, v_3)$. 可以看出, σ 就间接地表示 $\sigma^\pi \notin \text{ST}$. 我们现在证明这个断言是正确的.

引理 37H 如果 ST 是和谐的, 那么 $\sigma^\pi \notin \text{ST}$.

证明 用反证法. 假设 σ^π 可以从 ST 的公理推出, 令 k 是这个推理 $\mathcal{G}$, 那么 $\langle r, r, k\rangle \in D$.

$$\therefore A_E \vdash \delta(\mathbf{S}^r\mathbf{0}, \mathbf{S}^r\mathbf{0}, \mathbf{S}^k\mathbf{0});$$

$$\therefore A_E \vdash \exists v_3\delta(\mathbf{S}^r\mathbf{0}, \mathbf{S}^r\mathbf{0}, v_3);$$

即

$$A_E \vdash \neg\sigma.$$

利用解释 π 我们得出，只要 ST 不和谐，那么 $\neg\sigma^\pi$ 就在 ST 中. 因此

$$\text{ST 是和谐的} \Rightarrow \sigma^\pi \notin \text{ST}. \qquad \blacksquare$$

和本书中的许多证明一样，上述证明是在非形式化数学中进行的. 但实际上所有这些证明都可以在 ST 中完成. 更本质地说，所有的工作都能在 ST 中进行. 可以想象，其实我们就是这样做的. 除了证明一个句子

$$\text{“ST 是和谐的} \Rightarrow \sigma^\pi \notin \text{ST”}.$$

之外，在集合论的形式语言中，我们还得到一个从某个句子的 ST 的公理推出：

$$(\text{Cons(ST)} \to \square).$$

这里 Cons(ST) 是将“ST 是和谐的”在集合论语言中的翻译 (以一个好的方法). 类似地，$\square$ 是“$\sigma^\pi \notin$ ST”的翻译. 但是，在集合论语言中，我们已经有了一个句子断言 $\sigma^\pi \notin$ ST，这个句子是 σ^π. 这就说明$\square$就是 (或可以证明在 ST 中等价于)σ^π. 因此，我们把

274

$$(\text{Cons(ST)} \to \sigma^\pi)$$

作为 ST 的一个定理.

现在，我们可以把 $\square$ 看作 σ^π. 通过上面的讨论，我们希望能使读者相信这至少是可以证明的，并且从它我们可以得到下面的结果：

集合论中的哥德尔第二不完全性定理 句子 Cons(ST) 不是 ST 的定理，除非 ST 不和谐.

证明 根据上面 (有道理) 的讨论，有

$$(\text{Cons(ST)} \to \sigma^\pi)$$

是 ST 的定理. 因此如果 Cons (ST) 是 ST 的定理，那么 σ^π 也是. 但根据引理 37H，如果 $\sigma^\pi \in$ ST，ST 就是不和谐的. $\blacksquare$

当然如果 ST 不和谐，那么每个句子，包括 Cons (ST)，都是定理. 因此在 ST 内对 Cons (ST) 的证明不能使人相信 ST 是和谐的. (相反地，由哥德尔第二定理却能得出 ST 是不和谐.) 但在哥德尔之前，人们曾经希望可以从比集合论公理弱一些的假设中证明 Cons (ST)，当然这些假设都被设成和谐的. 现在我们知道 Cons(ST) 并不包含在 ST 的任何一个子理论中，除非 ST 是不和谐的.

现在我们知道了集合的任意一个递归可公理化的理论 (只要符合和谐的条件并且能推出现在已知的结论) 都不是完全理论. 这就提出了一个挑战：给这些理论增加公理. 一方面，我们要通过增加理论的公理，使得理论朝着我们认为有用的方向增强. 另一方面，希望增加的公理能使我们对集合的非形式化理解更准确，比如，集合到底是什么？它们有怎样的性质？

习题

1. 设 σ 是一个句子满足 PA $\vdash (\sigma \leftrightarrow \text{Prb}_{\text{PA}}\sigma)$. (即 σ 表示“我是可证明的. ”与“我是不可证明的”相反，这个句子也有这个有趣的性质.) 是否有 PA $\vdash \sigma$?

275

2. 设 T 是有限递归语言中的理论，并且假设存在一个 CnA_E 到 T 的解释. 证明 T 是强不可判定的，即，只要理论 T' 满足 $T \cup T'$ 是和谐的，那么 $\sharp T'$ 不是递归的.

3.8 幂乘运算的表示 [1]

在 3.1 节和 3.2 节中，我们学习了 $\mathfrak{N}$ 的一些归约理论，并证明了它们的可判定性. 在节 3.3 中，我们向语言中添加了乘法和幂乘运算. 得到的理论 (在 3.5 节中) 是不可判定的. 实际上，如果我们只添加乘法 (放弃幂乘)，仍然有这样的结果.

令 $\mathfrak{N}$ 去掉幂乘运算后得到的 $\mathfrak{N}$ 的归约模型为 $\mathfrak{N}_M$：

$$\mathfrak{N}_M = (\mathbb{N}; 0, S, <, +, \cdot).$$

符号 $\mathbf{E}$ 不出现在 $\mathfrak{N}_M$ 的语言中. 令 A_E 去掉 E_1 和 E_2 后得到的集合为 A_M. 这一节的目的是为了证明当我们用"A_M"和"$\mathfrak{N}_M$"代替"A_E"和"$\mathfrak{N}$"时，3.3 节至 3.5 节中的所有定理仍然成立. 那么需要证明的关键是幂乘运算在 $\mathrm{Cn}A_M$ 中是可表示的. 即，在 $\mathfrak{N}_M$ 的语言中存在公式 ε，使得对于任意的 a 和 b，

$$A_M \vdash \forall z[\varepsilon(\mathbf{S}^a\mathbf{0}, \mathbf{S}^b\mathbf{0}, z) \leftrightarrow z = \mathbf{S}^{(a^b)}\mathbf{0}].$$

这样 $\varepsilon(x, y, z)$ 就可以代替 $x\mathbf{E}y = z$，而不用使用符号 $\mathbf{E}$ 了.

如果我们想知道哪些关系和函数在 $\mathrm{Cn}A_M$ 中是可表示的，我们首先会发现除了幂乘运算外，所有在 $\mathrm{Cn}A_E$ 中可表示的在 $\mathrm{Cn}A_M$ 中都可表示. 也就是，3.3 节中到编目 7 之前列出的结果都成立. 因此，必须证明幂乘运算在 $\mathrm{Cn}A_M$ 中也是可表示的.

我们知道幂乘运算可以用下面的递归方程来刻画

$$a^0 = 1,$$

276

$$a^{b+1} = a^b \cdot a.$$

从我们对原始递归的了解 (3.3 节中的编目 13 加上习题 8)，我们希望定义

$$E^*(a,b) = 使得\ [(s)_0 = 1\ 并且对于所有 i < b,\ (s)_{i+1} = (s)_i \cdot a] 成立的最小的\ s$$

这样 $a^b = (E^*(a,b))_b$. 但这并不能得到我们想要的可表示性的证明，因为我们并不知道分解函数 $(a)_b$ 在 $\mathrm{Cn}A_M$ 中是可表示的. 实际上，我们并不真的需要这种分解函数 (它对应于编码序列的一种特殊方式). 我们所需要的是某种类似分解函数的函数 δ，下面的引理总结了它的性质.

引理 38A 存在一个在 $\mathrm{Cn}A_M$ 中可表示的函数 δ，使得对于每个 $n, a_1, \cdots, a_n$，存在一个 s，对于所有的 $i \leqslant n$，都有$\delta(s,i) = a_i$.

一旦这个引理被证明了，就能定义

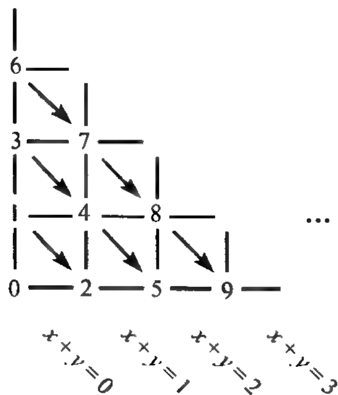
$$E^{**}(a,b) = 使得[\delta(s,0) = 1 并且对于所有 i < b,\ \delta(s,i+1) = \delta(s,i) \cdot a] 成立的最小的 s$$

这个引理保证满足条件的 s 的存在性. 那么 E^{**} 在 $\mathrm{Cn}A_M$ 中是可表示的，进而幂乘也是，因为 $a^b = \delta(E^{**}(a,b), b)$. 数论中的一些事实保证了引理中的函数 δ 是存在的.

1. 本节可以略过，并不影响本书的连续性.

3.8.1 配对函数

为了证明略去的引理，首先我们为编码数和解码数组成的数对构造一个函数. 众所周知，存在把 $\mathbb{N}\times\mathbb{N}$ 一对一映射到 $\mathbb{N}$ 上的函数. 特别地，下图中的函数 J 也是这样，坐标 $\langle a,b\rangle$ 处写的是 $J(a,b)$ 的值.



277

例如，$J(2,1)=8$，$J(0,2)=3$. 为了得到 $J(\mathrm{a},\mathrm{b})$ 的表达式，我们注意到，沿着 $x+y=n$ 这一行有 $n+1$ 个点 (在 $\mathbb{N}$ 中的坐标). 这样，

$$\begin{aligned}J(a,b)&=J\text{在平面上安排的较少的点数}\\&=[\text{对于}n=0,1,\cdots,(a+b-1),\ x+y=n\\&\qquad\text{行上的点数}]+[x+y=a+b\text{行上}x<a\text{的点数}]\\&=[1+2+\cdots+(a+b)]+a\\&=\frac{1}{2}(a+b)(a+b+1)+a\\&=\frac{1}{2}[(a+b)^2+3a+b].\end{aligned}$$

令 K 和 L 是 J 在坐标轴上的投影函数，即满足

$$K(J(a,b))=a,\ L(J(a,b))=b.$$

的唯一函数. 例如，$K(7)=1$，1 是 J 在平面上安排 7 的坐标点 $\langle 1,2\rangle$ 的 x 坐标. 类似地，$L(7)=2$，等于这个点的 y 坐标.

我们断言，J，K 和 L 在 $\mathrm{Cn}A_M$ 中都是可表示的. 函数

$$H(a)=\text{使得}a\leqslant 2b\text{的最小的 }b$$

对于任意偶数 a，有性质 $H(a)=\frac{1}{2}a$. 那么有，

$$\begin{aligned}J(a,b)&=H((a+b)\cdot(a+b+1))+a,\\K(p)&=\text{使得 [对于某个}b\leqslant p,J(a,b)=p]\text{的最小的 }a\\L(p)&=\text{使得 [对于某个 }a\leqslant p,J(a,b)=p]\text{的最小的 }b\end{aligned}$$

从上面的 4 个式子我们能断定 H，J，K 和 L 在 $\mathrm{Cn}A_M$ 中是可表示的.

3.8.2　哥德尔 β 函数

设 β 为如下定义的函数：

$$\begin{aligned}\beta(c,d,i) &= c \div [1+(i+1)\cdot d]\text{的余数} \\ &= \text{对于某个}q \leqslant c \\ &\qquad c = q\cdot[1+(i+1)\cdot d]+r\text{的最小的}r\end{aligned}$$

这个不太直观的函数却能很好地构造出引理 38A 中的分解函数. 令

$$\delta(s,i) = \beta(K(s), L(s), i).$$

显然，δ 在 $\mathrm{Cn}A_M$ 中是可表示的. 但并不显然的是，它符合引理 38A 中的条件. 我们要证明：

$$\begin{aligned}&\text{对于任意 } n \text{ 和任意} a_0,\cdots,a_n, \\ &\text{存在 } c \text{ 和 } d \text{ 使得对于所有} i \leqslant n, \beta(c,d,i) = a_i \qquad (*)\end{aligned}$$

278 再证明，对于所有 $i \leqslant n, \delta(J(c,d),i) = \beta(c,d,i) = a_i$.

式 (*) 是数论方面的句子，而不是逻辑方面的. 它的证明是以中国剩余定理为基础的. 称 $d_0,\cdots,d_n$ 是 *两两互素的* 整数，当且仅当对于任意的 $i \neq j$, d_i 和 d_j 没有共同的素因子.

中国剩余定理　设 $d_0,\cdots,d_n$ 是两两互素的整数，$a_0,\cdots,a_n$ 是自然数满足 $a_i < d_i$，那么能够找到一个数 c 使得对于所有 $i \leqslant n$,

$$a_i = c \div d_i\text{的余数}$$

证明　令 $p = \Pi_{i\leqslant n} d_i$，并且对于任意的 c，令 $F(c)$ 为 c 分别除以 $d_0,\cdots,d_n$ 所得的 $(n+1)$ 元余数组. 请注意，这个 $(n+1)$ 元余数组有 p 种可能的值.

我们说 F 在 $\{k|0 \leqslant k < p\}$ 上是一对一的. 因为假设 $F(c_1) = F(c_2)$，那么每个 d_i 都能整除 $|c_1 - c_2|$. 由于 d_i 是互素的，因此 p 一定能整除 $|c_1 - c_2|$. 又因为 c_1, c_2 比 p 小，所以必然有 $c_1 = c_2$.

这样 F 在 $\{k|0 \leqslant k < p\}$ 上的限制取遍所有的 p 种可能的值. 特别地，假设在某点 c 上的取值为 $\langle a_0,\cdots,a_n\rangle$. 这就是我们想要的 c. ■

引理 38B　对于任意的 $s \geqslant 0$，下面的 $s+1$ 个数：

$$1+1\cdot s!, 1+2\cdot s!, \cdots, 1+(s+1)\cdot s!$$

是两两互素的.

证明　所有这些数有一个共同的性质：任意一个素因子 q 不能整除 $s!$，因此 $q > s$. 如果素数 q 能整除 $1+j\cdot s!$ 和 $1+k\cdot s!$，那么它就能整除它们的差 $|j-k|\cdot s!$. 由于 q 不能整除 $s!$，因此它能整除 $|j-k|$. 但 $|j-k| \leqslant s < q$，所以只有当 $|j-k| = 0$ 时，这个式子才成立. ■

式(*) 的证明　假设已知 $a_0,\cdots,a_n$，我们必须找出 c 和 d，使得对于所有的 $i \leqslant n$，c 除以 $1+(i+1)\cdot d$ 的余数是 a_i.

令 s 是 $\{n, a_0,\cdots,a_n\}$ 中最大的一个，并另设 $d = s!$. 那么根据引理 38B，对于 $i \leqslant n$, $1+(i+1)\cdot d$ 是两两互素的. 因此由中国剩余定理知，存在一个数 c 使得对于所有的 $i \leqslant n$,

$c \div [1+(i+1)\cdot d]$ 的余数是 a_i. ■

到此，我们完成了引理 38A 的证明. 由引理后的讨论我们得到：

定理 38C 幂乘运算在 $\mathrm{Cn}A_M$ 中是可表示的.

有了这个定理，我们就能证明 3.3 节中的编目 7. 当时的证明表明了，编目 7 中的函数 (在 n 的值为 p_n) 在 $\mathrm{Cn}A_M$ 中是可表示的. 因为这个函数是由已知在 $\mathrm{Cn}A_M$ 中可表示的关系和函数 (包括幂乘) 通过允许的方式形成的. 279

同样的现象贯穿于整个 3.3 节和 3.4 节中. 当时给出的证明同样证明了 $\mathrm{Cn}A_M$ 中的可表示性. 因此任意递归关系在 $\mathrm{Cn}A_M$ 中是可表示的，并且如果这个关系恰好是个函数，那么它是函数可表示的. 3.5 节中用在 $\mathfrak{N}_M$ 和 A_M 上的证明同样可以用在 $\mathfrak{N}$ 和 A_E 上. 特别地，我们还得到 $\mathrm{Cn}A_M$ 的强不可判定性：在 $\mathfrak{N}$ 的语言中的任意理论 T，如果满足 $T \cup A_M$ 是和谐的，那么 T 就不是递归的.

请注意，任意在 $\mathfrak{N}$ 中可定义的关系 (即，算术关系) 在 $\mathfrak{N}_M$ 中也是可定义的. 对于幂乘运算来说，如果它在 $\mathrm{Th}\mathfrak{N}_M$ 的某个子理论中可表示，那么它在 $\mathfrak{N}_M$ 中更是可定义的. 根据新的塔斯基定理，$\sharp\mathrm{Th}\mathfrak{N}_M$ 在 $\mathfrak{N}_M$ 中是不可定义的，进而 $\sharp\mathrm{Th}\mathfrak{N}_M$ 不是算术的.

在 2.7 节中的技巧中，我们可以说存在一个 $\mathrm{Th}\mathfrak{N}$ 到 $\mathrm{Th}\mathfrak{N}_M$ 中的可靠解释. 它就是在 (除了 **E** 外的) 所有参数上的恒等解释，并且给 **E** 指派一个在 $\mathfrak{N}_M$ 中定义幂乘运算的公式.

表 3-2 中总结了第 3 章中有关数论及其归约的一些结果.

表 3-2

结构	理论	理论的模型	可定义集	注释
$(\mathbb{N})$	可判定, 不可有限公理化, 实现量词消去	任意无限集合	$\varnothing$ 和 $\mathbb{N}\{0\}$是不可定义的	
$(\mathbb{N};0)$	同上	含有不同元素的无限集	$\varnothing,\{0\},\mathbb{N}-\{0\},\mathbb{N},S$ 是不可定义的	
$(\mathbb{N};0,S)$	同上	标准部分加上 Z 链中的任意元素	有限集和余有限的集合 $<$ 是不可定义的	$\{0\}$ 在 $(\mathbb{N};S)$ 中是可定义的
$(\mathbb{N};0,S,<)$	可判定, 可有限公理化, 实现量词消去	同上, Z 链间可以有任意的序	有限集和余有限集序关系 $+$ 是不可定义的	$\{0\}$ 和 S 在 $(\mathbb{N};<)$ 中是可定义的
$(\mathbb{N};0,S,<,+)$	可判定 (Presburger)	Z 链满足无端点稠密序. 并且存在一个合适的加法运算	终周期集 $\cdot$ 是不可定义的	$\{0\}$, S 和 $<$ 在 $(\mathbb{N};+)$ 中是可定义的
$(\mathbb{N};0,S,<,+,\cdot)$	非算术的 $\therefore$ 不是递归可公理化的	同上, 但存在一个合适的乘法运算	所有的算术关系式都是可定义的	算术关系在 $(\mathbb{N};S,\cdot)$、$(\mathbb{N};+,\cdot)$ 和 $(\mathbb{N};<,D)$ 中是可定义的, 其中 $D(x,y)=(x)_y$

习题

1. 设 $D(a,b)=(a)_b$. 证明任意算术关系在结构 $(\mathbb{N};<,D)$ 中可定义. 注：有人可能会有疑问，为什么 $\mathrm{Th}\mathfrak{N}_A$，含有加法，是可判定的(3.2 节中证明)，而 $\mathrm{Th}\mathfrak{N}_M$，含有加法和乘法，却是不可判定的？答案是，正如本章所证明的，乘法能产生某些编码和解码序列. 这个习题的目的就是为了证明，一旦

我们有了解码函数 D 和序关系，就必然有加法，乘法和幂乘构成的充分复杂的算术.

2. 证明加法关系 $\{\langle a,b,c\rangle | a+b=c\}$ 在结构 $(\mathbb{N}; S, \cdot)$ 中是可定义的. 提示：在什么情况下等式 $S(ac)\cdot S(bc) = S(c\cdot c\cdot S(ab))$ 成立?

280～281

3. (a) 证明 $\mathrm{Th}(\mathbb{Z}; +, \cdot)$ 是强不可判定的. (见 3.7 节的习题 2.)

   (b) (假设读者有代数背景知识.) 证明环的理论是不可判定的, 并且交换环的理论也是不可判定的.

第4章

二 阶 逻 辑

4.1 二阶语言

为了得到比一阶语言更丰富，更富有表现力的语言，我们可以对谓词符号和函数符号添加量词. 例如，

$$\exists x(Px \to \forall xPx)$$

是一个含有参量 $\forall$ 和 P 的恒真式. 我们把

$$\forall P\exists x(Px \to \forall xPx)$$

也称为恒真式，因为无论 P 的解释是什么，它的取值都是真的. (此处的参量只有 $\forall$，P 被看作谓词变元.)

除了 2.1 节开始介绍的符号外，我们假设还有以下的逻辑符号：

(4) 谓词变元：对于每个正整数 n，我们有 n 元谓词变元

$$\boldsymbol{X}_1^n, \boldsymbol{X}_2^n, \cdots$$

(5) 函数变元：对于每个正整数 n，我们有 n 元函数变元

$$\boldsymbol{F}_1^n, \boldsymbol{F}_2^n, \cdots$$

为了防止混淆，以后把通常意义下的变元 $v_1, v_2, \cdots$ 称为 个体 变元. 项的定义和以前一样，是由常量符号和个体变元通过函数符号 (包括函数参量和函数变元) 形成的表达式. 原子公式仍然表示为 $Pt_1 \cdots t_n$，其中 $t_1, \cdots, t_n$ 是项，P 是 n 元谓词符号 (参量或变元). 合式公式的定义增加了新的构成运算：如果 φ 是合式公式，那么 $\forall \boldsymbol{X}_i^n \varphi$ 和 $\forall \boldsymbol{F}_i^n \varphi$ 也是. φ 中变元自由出现的概念和以前定义的相同. 句子是指不含有自由变元 (个体变元、谓词和函数变元) 的合式公式.

282

值得注意的是，谓词参量和自由谓词变元所起的作用在本质上是相同的. 类似地，函数参量和自由函数变元，常量符号和自由个体变元之间也有紧密的联系.

至于 结构，我们仍然是指满足 2.2 节中条件的参量集合上的函数. 但可满足性的定义必须自然地扩展. 设 V 是所有个体变元、谓词变元和函数变元的集合，并设 s 是 V 上的函数，它给每个变元分配一类合适的对象. 这样 $s(v_1)$ 就是论域中的一个元素，$s(\boldsymbol{X}^n)$ 是论域中的 n 元关系，$s(\boldsymbol{F}^n)$ 是 n 元运算. 对于项 t，$\bar{s}(t)$ 自然就有定义. 特别地，如果 $\boldsymbol{F}$ 是函数变元，那么 $\bar{s}(\boldsymbol{F}t_1 \cdots t_n)$ 就是函数 $s(\boldsymbol{F})$ 在 $\langle \bar{s}(t_1), \cdots, \langle \bar{s}(t_n) \rangle$ 上的值. 原子公式可满足的定义本质上与前面的定义相同. 对于一个谓词变元 $\boldsymbol{X}$，

$$\models_{\mathfrak{A}} \boldsymbol{X} t_1 \cdots t_n[s] \quad \text{iff} \quad \langle \bar{s}(t_1), \cdots, \bar{s}(t_n) \rangle \in s(\boldsymbol{X}).$$

可满足性定义中唯一新的特性是由新量词引出的.

(5) $\models_{\mathfrak{A}} \forall \boldsymbol{X}_i^n \varphi[s]$ 当且仅当对于 $|\mathfrak{A}|$ 上的每个 n 元关系 R, 有 $\models_{\mathfrak{A}} \varphi[s(X_i^n)|R]$.

(6) $\models_{\mathfrak{A}} \forall \boldsymbol{F}_i^n \varphi[s]$ 当且仅当对于每个 n 元函数 $f: |\mathfrak{A}|^n \to |\mathfrak{A}|$, 有 $\models_{\mathfrak{A}} \varphi[s(\boldsymbol{F}_i^n)|f]$.

显然, s 在公式中自由出现的变元上的值是有意义的. 对于句子 σ, 我们可以清楚地判断出它在 $\mathfrak{A}$ 中的取值是真还是假. 逻辑 (语义) 蕴涵的概念不变.

例 1 良序是一个序关系, 并且满足任意非空集合 (按照这个序关系) 都有一个最小元. 这个条件可以翻译成二阶句子

$$\forall \boldsymbol{X}(\exists y \boldsymbol{X} y \to \exists y(\boldsymbol{X} y \wedge \forall z(\boldsymbol{X} z \to y \leqslant z))).$$

这里, 和别处一样, 如果 $\boldsymbol{X}$ 和 $\boldsymbol{F}$ 没有下标, 我们可以省略不写. 如果上标可以从上下文中得出, 我们也可以省略.

283

例 2 皮亚诺代数的一条假设 (归纳假设) 是, 自然数的一个集合如果包含 0 并且对后继函数是封闭的, 那么这个集合实际上包含了所有的自然数. 这个假设也可以用数论的二阶语言写出来

$$\forall \boldsymbol{X}(\boldsymbol{X}\mathbf{0} \wedge \forall y(\boldsymbol{X} y \to \boldsymbol{X}\mathbf{S} y) \to \forall y \boldsymbol{X} y).$$

满足 S1, S2 和上述的皮亚诺归纳假设的任意一个模型都同构于 $(N; 0, S)$, 证明见习题 1. 因此这些句子的集合是范畴的, 即它的所有模型是同构的.

例 3 假设在公式 φ 中谓词变元 $\boldsymbol{X}^n$ 不是自由出现的, 那么公式

$$\exists \boldsymbol{X}^n \forall v_1 \cdots \forall v_n [\boldsymbol{X}^n v_1 \cdots v_n \leftrightarrow \varphi].$$

是恒真的. (此处 φ 中除了 $v_1, \cdots, v_n$ 之外, 其他的变元可以自由出现.) 这个公式表示存在一个 n 元关系满足 φ. 我们把这种形式的公式称作 *关系概括公式*, 类似地, 我们还有 *函数概括公式*. 如果在公式 ψ 中变元 $\boldsymbol{F}^n$ 不是自由出现的, 那么

$$\begin{aligned} &\forall v_1 \cdots \forall v_n \exists! v_{n+1} \psi \to \\ &\quad \exists \boldsymbol{F}^n \forall v_1 \cdots \forall v_{n+1} (\boldsymbol{F}^n v_1 \cdots v_n = v_{n+1} \leftrightarrow \psi) \end{aligned}$$

是恒真的. (这里 “$\exists! v_{n+1} \psi$” 是 2.2 节的习题 21 中得到的公式的缩写.)

例 4 在有序的实数域中, 任意有界的非空集合都有上确界. 我们能够用二阶句子表示这个句子:

$$\begin{aligned} &\forall \boldsymbol{X}[\exists y \forall z(\boldsymbol{X} z \to z \leqslant y) \wedge \exists z \boldsymbol{X} z \to \\ &\quad \exists y \forall y'(\forall z(\boldsymbol{X} z \to z \leqslant y') \leftrightarrow y \leqslant y')]. \end{aligned}$$

我们知道任意满足这个二阶句子的有序域都同构于有序实数域.

例 5 对于每个 $n \geqslant 2$, 我们都有一个一阶句子 λ_n 表示 “至少存在 n 个对象. ” 例如, λ_3 是

$$\exists x \exists y \exists z(x \neq y \wedge x \neq z \wedge y \neq z).$$

集合 $\{\lambda_2, \lambda_3, \cdots\}$ 表明它的结构的类 EC_Δ 包含无限结构. 我们说存在一个二阶句子与其等价. 一个集合是无限的, 当且仅当在这个集合上存在一个序, 在这个序下没有最后一个元素. 更简单地说, 一个集合是无限的, 当且仅当在这个集合上存在一个不自反的传递关系 R. 这个条件可以写成二阶句子 λ_∞:

284

$$\exists \boldsymbol{X}[\forall u \forall v \forall w(\boldsymbol{X}uv \to \boldsymbol{X}vw \to \boldsymbol{X}uw) \land \forall u \neg\, \boldsymbol{X}uu \land \forall u \exists v \boldsymbol{X}uv].$$

另一个定义无限结构的类的句子 (使用函数变元) 是

$$\exists \boldsymbol{F}[\forall x \forall y(\boldsymbol{F}x = \boldsymbol{F}y \to x = y) \land \exists z \forall x \boldsymbol{F}x \neq z],$$

它表示存在一个一对一但不是到上的函数.

上面这个例子说明紧致性定理在二阶逻辑中不成立:

定理 41A 存在一个不可满足的二阶句子集, 但它的每个有限子集都是可满足的.

证明 定理中的集合可以由上面例子中定义过的公式组成,

$$\{\neg\, \lambda_\infty, \lambda_2, \lambda_3, \cdots\} \qquad \blacksquare$$

洛文海 - 斯科伦定理在二阶逻辑中同样不成立. 所谓 *等号的语言* 是指除了 $\forall$ 之外不含其他参量的语言 (含有 $=$). 这个语言的结构可以简单地看作一个非空集合. 特别地, 从等价的意义讲, 结构是由它的基数决定的. 因此, 从逻辑等价的观点看, 这个语言中的句子是由它的模型的基数集合 (称为 *谱*) 决定的.

定理 41B 在有等号的二阶语言中存在一个句子, 它在一个集合中取值为真当且仅当集合的基数是 $2^{\aleph_0}$.

证明 (利用代数和分析中的概念) 首先考虑有序域 (一阶) 公理的合取式, 再把它与表示上确界的二阶句子合取 (见本节的例 4). 这个句子的模型正好是有序实数域的同构 (即结构同构于有序实数域). 现在我们把参量 $\mathbf{0}, \mathbf{1}, +, \cdot, <$ 看成适当的变元 (个体变元, 函数变元或谓词变元), 而只保留等号, 得到的句子就有所要的性质. $\blacksquare$

除了定理中的基数外, 还存在其他的基数具有这种二阶特性. 参见习题 2.

285

定理 41C 恒真的二阶句子的哥德尔数集在 $\mathfrak{N}$ 中不可能被任意的二阶公式定义.

在此, 我们假设哥德尔数已经通过以前使用的方法分配给了二阶公式. 尽管我们是在数论的二阶语言中展开证明, 但是这个定理对于任意至少含有一个二元谓词符号的有限递归语言都成立.

证明 设 T^2 是 $\mathfrak{N}$ 的二阶理论, 即在 $\mathfrak{N}$ 中真的二阶句子集. 使用证明塔斯基定理时所用的方法, 我们可以证明 $\sharp T^2$ 在 $\mathfrak{N}$ 中不可能被任意的二阶公式定义.

现在设 α 是 A_E(含有例子 2 中的二阶皮亚诺归纳假设) 中元素的合取式, 那么 α 的任意模型同构于 $\mathfrak{N}$, 参见习题 1. 因此, 对于任意句子 σ,

$$\sigma \in T^2 \quad \text{iff} \quad (\alpha \to \sigma)\text{是恒真的}.$$

进而, 恒真式 (的哥德尔数) 的集合是不能定义的, 除非 $\sharp T^2$ 能定义. $\blacksquare$

不仅如此，二阶恒真式的哥德尔数集不是算术的，并且不是递归可枚举的．也就是说，可枚举定理在二阶逻辑中也不成立．(我们可以证明这个集合在三阶，甚至 ω 阶的数论中也不能定义．但在此我们不讨论这些问题．)

如果比较二阶全称句子和对应的一阶形式，我们会发现一些有趣的结果．比如二阶皮亚诺归纳假设

$$\forall \boldsymbol{X}(\boldsymbol{X}\mathbf{0} \wedge \forall y(\boldsymbol{X}y \to \boldsymbol{X}\mathbf{S}y) \to \forall y\boldsymbol{X}y)$$

和相应的一阶公式集，也就是所有的句子集

$$\varphi(\mathbf{0}) \wedge \forall y(\varphi(y) \to \varphi(\mathbf{S}y)) \to \forall y\varphi(y),$$

其中 φ 是只有 v_1 为自由变元的一阶公式．如果 $\mathfrak{A}$ 是皮亚诺归纳假设的模型，那么 $|\mathfrak{A}|$ 的任意子集如果包含 $\mathbf{0}^{\mathfrak{A}}$ 并且在 $\mathbf{S}^{\mathfrak{A}}$ 下封闭，那么这个子集实际就是 $|\mathfrak{A}|$．另一方面，如果 $\mathfrak{A}$ 是相应原子公式的模型，那么说 $|\mathfrak{A}|$ 的每个可定义子集，只要包含 $\mathbf{0}^{\mathfrak{A}}$ 并且在 $\mathbf{S}^{\mathfrak{A}}$ 下是封闭的，就是 $|\mathfrak{A}|$ 极有可能存在一个不可定义的子集，使得这不成立．(例如，取 $\text{Th}(\mathbb{N}; 0, S)$ 的任意一个含有 Z 链的模型 $\mathfrak{A}$，那么 $\mathfrak{A}$ 满足上面的一阶公式，但不满足二阶归纳假设．标准点的集合仅仅在 $\mathfrak{A}$ 中不可定义．)

286

习题

1. 证明：含有 $\forall$、$\mathbf{0}$ 和 $\mathbf{S}$ 的语言的结构如果满足句子

$$\forall x\mathbf{S}x \neq \mathbf{0} \tag{S1}$$

$$\forall x\forall y(\mathbf{S}x = \mathbf{S}y \to x = y) \tag{S2}$$

和皮亚诺归纳假设

$$\forall \boldsymbol{X}(\boldsymbol{X}\mathbf{0} \wedge \forall y(\boldsymbol{X}y \to \boldsymbol{X}\mathbf{S}y) \to \forall y\boldsymbol{X}y)$$

那么它同构于 $\mathfrak{N}_S = (\mathbb{N}; 0, S)$.

2. (a) 请给出一个等于的二阶语言中的句子，使得它在一个集合中取值为真当且仅当集合的基数是 $\aleph_0$.

   (b) 对于 $\aleph_1$ 回答 (a) 中同样的问题.

3. 设 φ 是公式，其中只有 $\boldsymbol{X}$ 是自由出现的 n 元谓词变元．称 $|\mathfrak{A}|$ 上的 n 元关系 R 在 $\mathfrak{A}$ 中由 φ 蕴涵定义，当且仅当在 R 对 $\boldsymbol{X}$ 的指派下，$\mathfrak{A}$ 能够满足 φ，而在其他关系对 $\boldsymbol{X}$ 的指派下，$\mathfrak{A}$ 不能满足 φ．证明在 $\mathfrak{N}$ 中为真的一阶句子的哥德尔数集 $\sharp\text{Th}\,\mathfrak{N}$ 可以在 $\mathfrak{N}$ 中由一个公式定义，这个公式中没有约束的谓词变元和函数变元．提示：写出真句子集必须满足的条件.

4. 考虑含有一元谓词符号 I 和 S 以及二元谓词符号 E 的语言 (包含等号)．请找一个二阶句子 σ 使得 (i) 如果 A 是满足 $A \cap \mathcal{P}A = \varnothing$ 的集合，并且如果 $|\mathfrak{A}| = A \cap \mathcal{P}A$, $I^{\mathfrak{A}} = A$, $S^{\mathfrak{A}} = \mathcal{P}A$, $E^{\mathfrak{A}} = \{\langle a, b\rangle | a \in b \subseteq A\}$，那么 $\mathfrak{A}$ 是 σ 的模型；(ii) σ 的每个模型都同构于 (i) 中描述的某一类．注：概括地说，σ 表示“$S = \mathcal{P}I$”.

4.2　斯科伦函数

对于任一给定的一阶公式，我们要证明如何找出一种方法来得到与其逻辑等价的前束二阶公式，这个公式具有下面的特殊形式：

存在量词	全称个体量词	无量词公式

287

这是一个前束公式，这个公式中所有的全称量词都是个体量词，紧接其后的是存在个体量词和函数量词组成的串.

一个最简单的例子是

$$\forall x \exists y \varphi(x, y) \models\!\!\!\!\!\dashv \exists \boldsymbol{F} \forall x \varphi(x, \boldsymbol{F} x).$$

"$\dashv$" 方向是显然的. 至于 "$\models$" 方向，考虑结构 $\mathfrak{A}$ 和满足 $\forall x \exists y \varphi(x, y)$ 的指派函数 s. 我们知道，对于任意的 $a \in |\mathfrak{A}|$，至少存在一个 $b \in |\mathfrak{A}|$，使得

$$\models_{\mathfrak{A}} \varphi(x, y)[s(x|a)(y|b)].$$

因此我们对于每个 a 选取一个这样的 b，就能够得到一个 $|\mathfrak{A}|$ 上的函数 f，使得 $f(a) = b$. (此处用到了选择公理.) 那么

$$\models_{\mathfrak{A}} \forall x \varphi(x, \boldsymbol{F} x)[s(\boldsymbol{F}|f)].$$

函数 f 称作公式 $\forall x \exists y \varphi$ 在结构 $\mathfrak{A}$ 中的**斯科伦(Skolem)函数**.

这种方法有着广泛的应用. 又例如，假设我们有公式

$$\exists y_1 \forall x_1 \exists y_2 \forall x_2 \forall x_3 \exists y_3 \psi(y_1, y_2, y_3).$$

(我们列出了自由变元 y_1, y_2, y_3，但 ψ 中可能还有其他的自由变元.) 公式的左边已经有存在量词 $\exists y_1$. 剩下的就是

$$\forall x_1 \exists y_2 \forall x_2 \forall x_3 \exists y_3 \psi(y_1, y_2, y_3).$$

这可以看作第一个例子的特例 (令 $\varphi(x_1, y_2) = \forall x_2 \forall x_3 \exists y_3 \psi(y_1, y_2, y_3)$). 和前面一样，它等价于

$$\exists \boldsymbol{F}_2 \forall x_1 \forall x_2 \forall x_3 \exists y_3 \psi(y_1, \boldsymbol{F}_2 x_1, y_3).$$

现在最左边是 $\exists y_1 \exists \boldsymbol{F}_2$，那么剩下的是

$$\forall x_1 \forall x_2 \forall x_3 \exists y_3 \psi(y_1, \boldsymbol{F}_2 x_1, y_3).$$

根据同样的理由，它逻辑等价于

$$\exists \boldsymbol{F}_3 \forall x_1 \forall x_2 \forall x_3 \psi(y_1, \boldsymbol{F}_2 x_1, \boldsymbol{F}_3 x_1 x_2 x_3),$$

其中 $\boldsymbol{F}_3$ 是三元函数变元. 因此最初的公式等价于

$$\exists y_1 \exists \boldsymbol{F}_2 \exists \boldsymbol{F}_3 \forall x_1 \forall x_2 \forall x_3 \psi(y_1, \boldsymbol{F}_2 x_1, \boldsymbol{F}_3 x_1 x_2 x_3).$$

对于无量词 ψ，这就是我们所需要的形式.

斯科伦范式定理 对于任意一阶公式，我们能够找到一个与其逻辑等价的二阶公式，这个二阶公式满足：

288

(a) 第一个字符 (有可能是空的) 是存在个体量词或存在函数量词，紧接着

(b) (有可能是空的) 全称个体量词，紧接着

(c) 一个无量词公式.

我们可以用归纳法给出形式化的证明，但前面这个例子已经为我们展示了一般的方法.

已知全称 ($\forall_1$) 公式是量词都是全称量词的一阶前束式：$\forall x_1 \forall x_2 \cdots \forall x_k \alpha$，其中 α 是无量词的. 类似地，存在 ($\exists_1$) 公式就是量词都是存在量词的一阶前束式.

推论 42A　对于任意一阶公式 φ，我们能够在包含函数符号的扩展语言中找到一个全称公式 θ，使得 φ 可满足当且仅当 θ 可满足.

把这个推论用到 $\neg\varphi$ 上，我们就能得到一个 (含有函数符号的) 存在公式，这个存在公式是恒真的当且仅当 φ 是恒真的.

证明　我们同样用例子来说明证明的过程. 设 φ 是下面的公式：

$$\exists y_1 \forall x_1 \exists y_2 \forall x_2 \forall x_3 \exists y_3 \psi(y_1, y_2, y_3).$$

首先，将 φ 用斯科伦形式的逻辑等价公式来代替

$$\exists y_1 \exists \boldsymbol{F}_2 \exists \boldsymbol{F}_3 \forall x_1 \forall x_2 \forall x_3 \psi(y_1, \boldsymbol{F}_2 x_1, \boldsymbol{F}_3 x_1 x_2 x_3).$$

那么对于 θ 我们取

$$\forall x_1 \forall x_2 \forall x_3 \psi(c, f x_1, g x_1 x_2 x_3),$$

其中 c, f 和 g 是新的函数符号，分别为零元，一元和三元. 一般说来，θ 并不逻辑等价于 φ. 但我们的确有 $\theta \vDash \varphi$(在扩展语言中). 并且 φ 的任一模型 $\mathfrak{A}$ 可以扩展成 θ 的模型 (通过正确地定义 $c^{\mathfrak{A}}, f^{\mathfrak{A}}$ 和 $g^{\mathfrak{A}}$). 这样 θ 和 φ 就是“等价可满足的”. ■

这个结果把检验一阶公式的可满足性这个一般问题简化为 (含有函数符号的) 全称公式的可满足性这个特殊情况. 同样地，它把恒真式的检验问题简化成 $\exists_1$ 的情况. 从这些简化过程，我们能够得出一阶逻辑的不可判定性的结果：

推论 42B　对于一个递归可数语言，语言中含有一个二元谓词符号，并且对于每个 $k \geqslant$ 289 0，都有无限多个 k 元函数符号.

(a) 可满足的全称句子的哥德尔数集不是递归的.

(b) 恒真的存在句子的哥德尔数集不是递归的.

证明　(b) 对于给定的任意句子 σ，将推论 42A 用到 $\neg\sigma$ 上，我们能够能行地找到一个存在句子，它是恒真的当且仅当 σ 是恒真的. 因此与丘奇定理相反，我们能够从存在恒真句子的判定过程得到任意恒真句子的判定过程. ■

在这些结果中，我们可以使用谓词变元而不使用函数变元，但必须付出很高的代价. 假设从一个一阶公式出发，它等价于具有斯科伦范式形式的公式 ψ，为了简便，我们设 $\psi = \exists \boldsymbol{F}\varphi$，其中 φ 中只含有个体量词，$\boldsymbol{F}$ 是一元函数变元. φ 可以这样选取，$\boldsymbol{F}$ 只出现在具有形式 $u = \boldsymbol{F}t$ 的方程中 (项 t 和 u 都不包含 $\boldsymbol{F}$)，这可以通过替换得到，例如，原子公式 $\alpha(\boldsymbol{F}t)$ 可以用 $\forall x(x = \boldsymbol{F}t \to \alpha(x))$ 或 $\exists x(x = \boldsymbol{F}t \wedge \alpha(x))$ 来代替.

接下来，我们会发现公式

$$\exists \boldsymbol{F} _ u = \boldsymbol{F}t _,$$

其中 $\boldsymbol{F}$ 只出现在上面提到的方程中，它等价于

$$\exists \boldsymbol{X}(\forall y \exists! z \boldsymbol{X} yz \wedge _\boldsymbol{X} tu_).$$

如果我们仔细研究这个问题 (在此我们不这样做) 就会发现，任意一个一阶公式都逻辑等价于一个二阶公式，这个二阶公式由下面几个部分组成：

(a) 存在谓词量词，紧接着

(b) 全称个体量词，紧接着

(c) 存在个体量词，紧接着

(d) 无量词公式.

推论 42A 和 42B 都有对应的类似情况，见习题 4. 与推论 42A 类似的情况把检验一阶公式的可满足性问题简化为 $\forall_2$ 公式 (含有谓词符号) 的特殊情况，检验恒真式的问题简化为 $\exists_2$ 公式的情况.

我们可以把推论 42B 类似的情况与 2.6 节中的习题 10 相比较，在习题 10 中已经证明了不含函数符号的 $\forall_2$ 恒真式集是可判定的.

Herbrand 扩展

在推论 42A 中，我们已经知道了如何找到与一阶公式“等价可满足”的全称公式. 并且一阶逻辑的可满足性问题可以归结为全称公式的可满足性问题. 290

现在再更进一步：从较弱的意义上说，全称公式的可满足性可以归结于 **命题** 逻辑的可满足性.

例 我们知道 $\forall x \exists y Pxy \not\models \exists y \forall x Pxy$. 但现在假设我们不知道这一点，而我们想知道这一逻辑推论是否成立. 也就是等价于要判定，假设 $\forall x \exists y Pxy$ 和结论的否定 $\neg \exists y \forall x Pxy$ 是否不可同时满足.

根据斯科伦范式定理，我们用某些逻辑等价句子来代替这些句子. 我们希望能够知道 $\exists \boldsymbol{F} \forall x Px\boldsymbol{F}x$ 和 $\exists \boldsymbol{G} \forall y \neg P\boldsymbol{G}yy$ 是否不可同时满足. 正如推论 42A 一样，用等价可满足的全称句子来代替这些句子，我们想知道集合 $\{\forall x Pxfx, \forall y \neg Pgyy\}$ 是否不可满足 (其中 f 和 g 是新的函数符号).

但这个全称句子的集合是可满足的，并且从这个集合可以造出它自己的模型. 下面就看看这个过程是怎么实现的. 我们用 Herbrand 域 H 作为模型的论域，所谓 Herbrand 域是指所有项 (在含有 f 和 g 的语言中) 的集合. 这样，对于每个变元 u，H 包含项

$$u, fu, gu, ffu, fgu, \cdots$$

令 Δ 是全称句子的 **实例** 组成的集合，也就是去掉全称句子中的全称量词，将全称量词变元换成 Herbrand 域中的任意一个项. 这样，对于每个变元 u, Δ 包含无量词公式

$$Pufu, Pgufgu, \cdots, \neg Pguu, \neg Pgfufu, \cdots$$

现在，我们从命题逻辑的观点来考虑 Δ. 命题符号是原子公式，如 $Pgfufu$ 是原子公式，并且这个例子中的 Δ 在命题逻辑中是可满足的. 也就是存在一个命题符号上的真值指派 v 使得对于 Δ 中的每个 α，都有 $\bar{v}(\alpha) = T$. 下面就是这样的一个 v：

$$v(Pt_1t_2)=\begin{cases} T & \text{如果 } t_1 \text{ 比 } t_2 \text{ 短} \\ F & \text{如果 } t_1 \text{ 不比 } t_2 \text{ 短} \end{cases}$$

最后，运用这个真值指派 v(在命题逻辑中) 得到的结构 $\mathfrak{H}$(一阶逻辑中的) 就是全称句子

291

的模型. 其中论域是 Herbrand 域：$|\mathfrak{H}|=H$. (有一个模仿 2.5 节中完全性定理的证明.)函数符号都可以用自身来解释：$f^{\mathfrak{H}}(t)$ 是 ft, $g^{\mathfrak{H}}(t)$ 是 gt. v 用来解释谓词符号 P：

$$\langle t_1,t_2\rangle\in P^{\mathfrak{H}}\Leftrightarrow v(Pt_1t_2)=T$$

这个结构就开始起作用了. 首先，$\models_{\mathfrak{H}} xPxfx$, 这是因为对于 Herbrand 域中的每个项 t, $\langle t,ft\rangle\in P^{\mathfrak{H}}$. 其次，$\models_{\mathfrak{H}}\forall y\neg Pgyy$, 这是因为对于 Herbrand 域中的每个项 t, $\langle gt,t\rangle\notin P^{\mathfrak{H}}$.

因此，得出的结论是，假设 $\forall x\exists yPxy$ 和结论的否定 $\neg\exists y\forall xPxy$ 实际上是可满足的，所以 $\forall x\exists yPxy\not\models\exists y\forall xPxy$.

我们能从这个例子得到什么？为了简便，我们假设语言中不含等号. 如果我们想知道对于集合 Γ 是否有 $\Gamma\models\varphi$, 其中 φ 是一阶逻辑中的公式. 这就等价于判断集合 $\Gamma;\neg\varphi$ 是否是不可满足的.

我们把这里的每个公式都用与其逻辑等价的斯科伦范式代替. 就像推论 42A 中的结论一样，我们能够得到一个等价可满足的全称公式集 Ψ. (在使用斯科伦范式时，用*不同的*斯科伦函数符号来表示每一个公式，因此，公式之间就不会有冲突了.) 进而有：

$$\Gamma\models\varphi\Leftrightarrow\quad\Psi\text{ 是不可满足的.}$$

Ψ 是全称公式的集合.

令 H 是 Herbrand 域，即在 Ψ 的语言中所有的项的集合，Δ 是 Ψ 中全称公式的实例集合. (即，去掉全称句子中的全称量词，将全称量词变元代换成 Herbrand 域中的任意一个项后得到的公式) 那么 Δ 只包含无量词公式. 我们从命题逻辑的角度来考虑 Δ, 其中命题符号是原子公式.

情形 I：Δ 在命题逻辑中不可满足. 在这种情况下，我们断定 Ψ 是不可满足的并且在一阶逻辑中 $\Gamma\models\varphi$, 这是因为全称公式逻辑蕴涵它的所有实例. 因此对于 Δ 中的每一公式 δ, $\Psi\models\delta$(在一阶逻辑中). Ψ 的任意模型一定是 Δ 的模型. 但是从 Δ 的模型 $\mathfrak{A}$ 中我们能够得到在命题逻辑中满足 Δ 的真值指派 v. (大家可以回忆一下 2.4 节中的习题 3, 一阶逻辑和

292

命题逻辑之间有趣的相互影响.)

情形 II：Δ 在命题逻辑中是可满足的，那么就有符合条件的真值指派 v. 我们可以利用 v 来构造结构 $\mathfrak{H}$ 使得 Ψ 是可满足的并且 $\Gamma\not\models\varphi$, 这是由于 $\mathfrak{H}$ 中有反例.

正如例子中所示，论域 $|\mathfrak{H}|$ 是 Herbrand 域 H, 即在 Ψ 的语言中所有项的集合. 函数符号也用其自身来解释：$f^{\mathfrak{H}}(t_1,\cdots,t_n)=ft_1\cdots t_n$. 我们用真值指派 v 来解释谓词符号 P：

$$\langle t_1,\cdots,t_n\rangle\in P^{\mathfrak{H}}\Leftrightarrow v(Pt_1\cdots t_n)=T$$

那么我们断言，Ψ 中的每个公式都在 $\mathfrak{H}$ 中被满足，只要将公式中的变元 x 由恒等函数 $s(x)=x$ 解释. 首先，对于 H 中的任意一个项 t, 有 $\bar{s}(t)=t$, 这和 2.5 节中完全性定理证明的第 4 步相同. 其次，对于所有的原子公式 $Pt_1\cdots t_n$,

$$\models_{\mathfrak{H}}Pt_1\cdots t_n[s]\Leftrightarrow\langle t_1,\cdots,t_n\rangle\in P^{\mathfrak{H}}\Leftrightarrow v(Pt_1\cdots t_n)=T$$

再由 2.4 节的习题 3, 我们可得 Δ 中的任意公式 δ 在 $\mathfrak{H}$ 中由 s 满足 (因为 $\bar{v}(\delta)=T$).

现在我们考虑 Ψ 中的任意公式, 它是全称公式, 为了简化此符号, 设它为 $\forall v_1\forall v_2\theta(v_1,v_2,v_3)$, 其中 θ 是无量词公式. 我们必须验证对于 H 中的任意项 t_1 和 t_2 有 $\vDash_{\mathfrak{H}}\theta[\![t_1,t_2,t_3]\!]$. 这等价于 (根据替换引理) 说公式 $\theta(t_1,t_2,v_3)$ 在 $\mathfrak{H}$ 中由 s 满足. 但这个公式是 $\forall v_1\forall v_2\theta(v_1,v_2,v_3)$ 的实例, 因此 $\theta(t_1,t_2,v_3)$ 在 Δ 中. 正如上面提到的, 我们的目的是为了证明 Δ 中的每个公式都在 $\mathfrak{H}$ 中由 s 满足, 这就是我们所要的.

下面的定理总结了我们的结果. 为了简便, 结果只涉及句子.

Herbrand 定理 在不含等号的一阶语言中, 我们考虑命题集 $\Gamma:\varphi$. Δ 如上所述. 那么要么 (情形 I)Δ 在命题逻辑中不可满足并且 $\Gamma\vDash\varphi$, 要么 (情形 II)Δ 在命题逻辑中可满足并且上面所构造的结构 $\mathfrak{H}$ 是 Γ 的一个模型, φ 在这个模型中是假的.

(Herbrand 在 1930 的论文中提出了这个工作, 不久之后, 他就在登山事故中去世了. 他对定理的叙述与上面这个定理完全不同, 但上面这个定理的思想来源于 Herbrand 和斯科伦在 1928 年的著作.)

在情形 I 中, 根据命题逻辑的紧致性定理, Δ 的某个有限子集是不可满足的. 通过这个事实, 我们可以不用 2.5 节或 2.4 节中的演绎算法就能证明一阶逻辑中的紧致性定理.

不仅如此, 可枚举性定理的证明也可以从 Herbrand 方法中得到, 而不用依靠 2.4 节和 2.5 节的结果. 现在取 $\Gamma\neq\varnothing$ 的特殊情况. 如果 φ 是恒真的, 那么随着 Δ 中元素的增加, 我们会得到一个不可满足的集合, 这个集合的不可满足性可以通过真值表证明. 如果 φ 不是恒真的, 那么随着 Δ 中元素的增加, 我们会得到一个结构, 在这个结构中 φ 不成立, 并且结构是无限的, 构造过程永远不会停止.

293

习题

1. 证明洛文海 – 斯科伦定理的推广形式: 设 $\mathfrak{A}$ 是可数语言的结构, S 是 $|\mathfrak{A}|$ 的可数子集. 那么存在 $\mathfrak{A}$ 的可数子结构 $\mathfrak{B}$, 满足 $S\subseteq|\mathfrak{B}|$ 并且对于将变元映射到 $|\mathfrak{B}|$ 中的任意函数 s 和任意 (一阶) 公式 φ, 有

$$\vDash_{\mathfrak{A}}\varphi[s]\quad\text{当且仅当}\quad\vDash_{\mathfrak{B}}\varphi[s].$$

提示: 为所有的公式选择斯科伦函数, 取 S 在函数下的闭集. *说明*: 具有这个性质的子结构 $\mathfrak{B}$ 称为*初等子结构*. 注意这个性质能推出 $\mathfrak{A}\equiv\mathfrak{B}$(取 φ 为句子). 一方面, 这个推广形式给出了比 2.6 节中的结果更强的结论. 我们不仅得到了 Th $\mathfrak{A}$ 的某个可数模型, 还得到了可数的子模型. 另一方面, 证明过程使用了选择公理.

2. 把上一题的结论推广到不可数的情形. 假设 $\mathfrak{A}$ 是基数为 λ 的语言的结构, S 是 $|\mathfrak{A}|$ 的子集, 基数为 κ. 证明存在 $\mathfrak{A}$ 的初等子结构 $\mathfrak{B}$, 它的基数至多为 $\kappa+\lambda$ 且 $S\subseteq|\mathfrak{B}|$.
3. 证明推论 42B 可以优化为以下情形:
   (a) 对于任意 $\exists_1$ 句子 σ, 我们能够能行地判定 σ 是否是可满足的.
   (b) 对于任意 $\forall_1$ 句子 σ, 我们能够能行地判定 σ 是否是恒真的.
4. (a) 写出本节末尾的两个推论 (与 42A 和 42B 类似).
   (b) 给出证明.
5. 还是 Herbrand 扩展中的例子, 但对反面: $\exists y\forall xPxy\vDash\forall x\exists yPxy$. 证明在这种情况下, 集合 Δ 在命题逻辑中是不可满足的.
6. 利用 Herbrand 扩展的方法证明: $\vDash\exists x(Px\to\forall xPx)$.
7. 修改 Herbrand 扩展的构造方法, 使其成为包含等号的语言. *提示*: 加上 2.5 节中证明完全性定理

294 的第 5 步，增加足够的存在句子以保证 $\{\langle t_1, t_2\rangle | v(t_1 = t_2) = T\}$ 是同余关系.

4.3 多类逻辑

现在我们回到一阶语言，但变元有许多类，包括不同的论域. (在下一节中，一类变元用来表示论域中的元素，另一类表示论域的子集，还有一类表示二元关系，等等.)

在非正式的数学中，有时会有这样的表述“我们用希腊字母表示序数，大写字母表示整数集合，……”为此，我们有效地使用多类变元，每一类都有自己的论域. 现在要仔细地检查这种情况，结果正如我们所期待的那样，所有结论和通常的一类情形相似，因此大部分证明都省略了.

假设有非空集合 I，它的元素称为类以及符号如下定义：

A. 逻辑符号

0. 括号：(,).
1. 命题联结符号：$\neg$, $\rightarrow$.
2. 变元：对于每个类 i，存在这个类的变元 $v_1^i, v_2^i, \cdots$.
3. 等于号：对于某些 $i \in I$，存在符号 $=_i$，表示类 $\langle i, i\rangle$ 的谓词符号.

B. 参量

0. 量词符号：对于每个类 i，存在一个全称量词符号 $\forall_i$.
1. 谓词符号：对于每个 $n > 0$ 和类的任意 n 元组 $\langle i_1, \cdots, i_n\rangle$，存在 n 元谓词符号的集合 (有可能是空的)，我们把这些谓词符号称为类 $\langle i_1, \cdots, i_n\rangle$ 的谓词符号.
2. 常量符号：对于每个类 i，存在着常量符号的集合 (有可能是空的)，我们把这些常量符号称为类 i 的谓词符号.
3. 函数符号：对于每个 $n>0$ 和类的 $n+1$ 元组 $\langle i_1, \cdots, i_n, i_{n+1}\rangle$，存在 n 元函数符号的集合 (有可能是空的)，我们把这些函数符号称为类 $\langle i_1, \cdots, i_n, i_{n+1}\rangle$ 的函数符号.

   和往常一样，我们必须假设这些符号的范畴是不交的，并且任意一个符号都不是其他符号的有限部分.

295 每个项也会被指派到唯一的一个类中. 对于所有的 i，我们递归地定义类 i 中的项集：

(1) 类 i 中的任意变元或常量符号是类 i 的项.

(2) 如果 $t_1, \cdots, t_n$ 分别是类 $i_1, \cdots, i_n$ 的项，f 是类 $\langle i_1, \cdots, i_n, i_{n+1}\rangle$ 的函数符号，那么 $ft_1 \cdots t_n$ 是类 i_{n+1} 的项.

这个定义可以推广到更普遍的形式. 由类 i 中的项 t 构成的序对 $\langle t, i\rangle$ 的集合是由基础集通过运算生成的. 这个基础集是

$$\{\langle v_n^i, i\rangle | n \geqslant 1 \ \&\ i \in I\} \cup \{\langle c, i\rangle | c \text{ 是类 } i \text{ 的常量符号}\}.$$

生成过程是这样的，对于类 $\langle i_1, \cdots, i_n, i_{n+1}\rangle$ 的函数符号 f，由序对 $\langle t_1, i_1\rangle, \cdots, \langle t_n, i_n\rangle$ 生成序对 $\langle ft_1 \cdots t_n, i_{n+1}\rangle$.

原子公式是符号序列 $Pt_1 \cdots t_n$，序列中分别包含类 $\langle i_1, \cdots, i_n\rangle$ 的谓词符号和类 $i_1, \cdots, i_n$ 的项 $t_1, \cdots, t_2$. 非原子公式就是用联结符号 $\neg$, $\rightarrow$ 和量词 $\forall_i v_n^i$ 形成的.

多类结构 $\mathfrak{A}$ 可以看作参量集上的函数，它给每个参量指派一个正确的对象类型：

(1) 对于量词符号 $\forall_i$, $\mathfrak{A}$ 指派一个非空集合 $|\mathfrak{A}|_i$, 称为类 i 的 $\mathfrak{A}$论域.

(2) 对于类 $\langle i_1, \cdots, i_n \rangle$ 的每个谓词 P, $\mathfrak{A}$ 指派一个关系

$$P^{\mathfrak{A}} \subseteq |\mathfrak{A}|_{i_1} \times \cdots \times |\mathfrak{A}|_{i_n}.$$

(3) 对于类 i 的每个常量符号 c, $\mathfrak{A}$ 指派 $|\mathfrak{A}|_i$ 中的一个点 $c^{\mathfrak{A}}$.

(4) 对于类 $\langle i_1, \cdots, i_n, i_{n+1} \rangle$ 中的每个函数符号 f, $\mathfrak{A}$ 指派一个函数

$$f^{\mathfrak{A}} : |\mathfrak{A}|_{i_1} \times \cdots \times |\mathfrak{A}|_{i_n} \to |\mathfrak{A}|_{i_{n+1}}.$$

真值和满足的定义也是显然的, $\forall_i$ 的意思是"对于类 i 的论域 $|\mathfrak{A}|_i$ 中的所有元素."

在多类结构中, 各种类的论域可以相交也可以不相交. 但由于类之间没有等于符号, 因此我们把相交看作特殊的情况. 特别地, 总存在与之初等等价的结构, 而此结构的论域是不交的.

归约到一类逻辑

多类语言在有些情形下使用比较方便, 但使用它和不使用它所得到的结果没有本质的不同. 这一节我们就将详细地讨论这个问题. 296

我们考虑一个一类语言, 这个语言中含有多类语言中所有的谓词, 常量和函数符号. 另外, 对于 I 中的每个 i, 它还含有一个一元谓词符号 Q_i. 按照句法翻译, 我们能把每个多类公式 φ 译成对应的一类公式 φ^*. 在翻译的过程中, 所有的等号都用 $=$ 代替, 其他的改变发生在量词 (量词符号和量词变元) 上: 我们用

$$\forall_i v^i_n __ v^i_n __,$$

替代

$$\forall v(Q_i v \to _v_)$$

其中变元 v 和其他变元不相同. 这样, 类 i 的量词就对应于 Q_i. (自由变元不变.)

现在考虑语义方面的问题. 我们同样能把多类结构 $\mathfrak{A}$ 转变为上述一类语言的结构 $\mathfrak{A}^*$. 论域 $|\mathfrak{A}^*|$ 是 $\bigcup_{i \in I} |\mathfrak{A}|_i$, $\mathfrak{A}$ 的所有论域的并. Q_i 对应于集合 $|\mathfrak{A}|_i$. $\mathfrak{A}^*$ 中的谓词和常量的解释和 $\mathfrak{A}$ 中相同. 对于函数符号 f, 函数 $f^{\mathfrak{A}^*}$ 是 $f^{\mathfrak{A}}$ 的任意扩充. (当然, 最后一个句子并不能完全确定 $f^{\mathfrak{A}^*}$. 我们针对 $\mathfrak{A}^*$ 给出的结果对于用上述方法得到的所有结构都是成立的.)

引理 43A 多类句子 σ 在 $\mathfrak{A}$ 中的取值为真, 当且仅当 σ^* 在 $\mathfrak{A}^*$ 中的取值为真.

为了证明这个引理, 我们必须给出一个有关公式的断言:

$$\vDash_{\mathfrak{A}} \varphi[s] \Longleftrightarrow \vDash_{\mathfrak{A}^*} \varphi^*[s]$$

其中 $s(v^i_n) \in |\mathfrak{A}|_i$. 这个断言将在后面用归纳法证明.

现在考虑另一个方向. 一类结构并不总能转化成多类结构, 因此必须加入一些条件. 设 Φ 是包含下列一类句子的集合:

(1) $\exists v Q_i v$, 对于 I 中的每个 i 成立.

(2) $\forall v_1 \forall v_n (Q_{i_1} v_1 \to \cdots \to Q_{i_n} v_n \to Q_{i_{n+1}} f v_1 \cdots v_n)$, 对于类 $\langle i_1, \cdots, i_n, i_{n+1} \rangle$ 的每个函数符号 f 成立. 我们把 $n = 0$ 的情况也包含进去, 当 $n = 0$ 时上述情况就变成 $Q_i c$, 对于类 i 的每个常量 c 成立.

297 请注意，上面的 $\mathfrak{A}^*$ 是 Φ 的一类模型. Φ 的一类模型 $\mathfrak{B}$ 确实能够转化成多类模型$\mathfrak{B}^\sharp$. 转化过程可以自然地进行：

$$|\mathfrak{B}^\sharp|_i = Q_i^{\mathfrak{B}};$$
$$P^{\mathfrak{B}^\sharp} = P^{\mathfrak{B}} \cap (Q_{i_1}^{\mathfrak{B}} \times \cdots \times Q_{i_n}^{\mathfrak{B}}), \text{其中 } P \text{ 是类 } \langle i_1, \cdots, i_n\rangle \text{ 的谓词符号}$$
$$c^{\mathfrak{B}^\sharp} = c^{\mathfrak{B}};$$
$$f^{\mathfrak{B}^\sharp} = f^{\mathfrak{B}} \cap (Q_{i_1}^{\mathfrak{B}} \times \cdots \times Q_{i_n}^{\mathfrak{B}} \times Q_{i_{n+1}}^{\mathfrak{B}}), \text{即 } f^{\mathfrak{B}} \text{ 在 } Q_{i_1}^{\mathfrak{B}} \times \cdots \times Q_{i_n}^{\mathfrak{B}} \text{ 上的限制,}$$
$$\text{其中 } f \text{ 是类 } \langle i_1, \cdots, i_2, i_{n+1}\rangle \text{ 的函数符号。}$$

引理 43B 如果 $\mathfrak{B}$ 是 Φ 的模型，那么 $\mathfrak{B}^\sharp$ 是多类结构，并且多类句子 σ 在 $\mathfrak{B}^\sharp$ 中的取值为真当且仅当 σ^* 在 $\mathfrak{B}$ 中的取值为真.

证明类似于引理 43A 的证明.

请注意，$\mathfrak{B}^{\sharp *}$ 一般情况下不等于 $\mathfrak{B}$. (例如，$|\mathfrak{B}|$ 可以包含不属于任意 $Q_i^{\mathfrak{B}}$ 的点.) 另一方面，$\mathfrak{A}^{*\sharp}$ 却等于 $\mathfrak{A}$.

定理 43C 在多类语言中，$\Sigma \vDash \sigma$ 当且仅当 $\mathfrak{B}$ 在一类语言中，$\Sigma^* \cup \Phi \vDash \sigma^*$.

证明 ($\Rightarrow$) 假设 $\Sigma \vDash \sigma$ 并且设 $\mathfrak{B}$ 是 $\Sigma^* \cup \Phi$(其中 $\Sigma^* = \{\sigma^* | \sigma \in \Sigma\}$) 的一类模型. 那么根据引理 43B, $\mathfrak{B}^\sharp$ 是 Σ 的模型，因此 $\mathfrak{B}^\sharp$ 是 σ 的模型. 所以再根据引理 43B, $\mathfrak{B}$ 是 σ^* 的模型.

($\Leftarrow$) 类似地，使用引理 43A. ■

使用定理 43C，我们可以把一类语言中的结果推广成下面的三个定理.

紧致性定理 如果多类句子集 Σ 的每个有限子集都有模型，那么 Σ 有模型.

证明 假设 Σ 的每个有限子集 Σ_0 都有一个多类模型 $\mathfrak{A}_0$，那么 Σ^* 的有限子集 Σ_0^* 有模型 $\mathfrak{A}_0^*$. 因此由一类语言的紧致性定理，Σ^* 有模型 $\mathfrak{B}$，那么 $\mathfrak{B}^\sharp$ 是 Σ 的一个模型. ■

可枚举性定理 在可数递归的多类语言中，恒真式的哥德尔数集是递归可枚举的.

证明 对于一个多类公式 σ，根据定理 43C 有

298
$$\vDash \sigma \quad \text{iff} \quad \Phi \vDash \sigma^*.$$

由于 Φ 是递归的，所以 $\text{Cn}\ \Phi$ 是递归可枚举的. 并且 σ^* 递归依赖于 σ，因此运用 3.5 节中的习题 7(b) 可以得到这个定理. ■

洛文海 – 斯科伦定理 对于可数语言中的任意多类结构，都存在着与之初等等价的可数结构.

证明 设给定的结构为 $\mathfrak{A}$，则 $\mathfrak{A}^*$ 是 $(\text{Th}\ \mathfrak{A})^* \cup \Phi$ 的一类模型. 根据通常的洛文海 - 斯科伦定理，$(\text{Th}\ \mathfrak{A})^* \cup \Phi$ 有可数模型 $\mathfrak{B}$. $\mathfrak{B}^\sharp$ 是 $\text{Th}\ \mathfrak{A}$ 的模型，因此初等等价于 $\mathfrak{A}$. ■

4.4 广义结构

现在我们回到在 4.1 节中开始讨论的二阶逻辑，我们讨论了二阶逻辑中的 (a) 语法，即二阶语言中的合式公式，和 (b) 语义，即结构 (和一阶语言中的一样) 的概念，满足和真值的定义。

在本节中，我们要在 (a) 不变的情况下，给出 (b) 的另一种形式. 简要地说：我们现在把 (原来认为的二阶) 语言看作多类的初等 (即一阶) 语言. 这样做是为了不仅对个体变元的论域给出解释，而且对谓词和函数变元的论域也给出解释. 在本节末尾大家将看到，这种方法特别适用于数论.

4.4.1 多类语言

就算不考虑我们的最终目标：4.1 节中的语法，研究由 4.1 节中的二阶语言构造的 (一阶) 多类语言也是很有用的. 我们取 $\aleph_0$ 个类：一个个体变元类 (变元为 $v_1, v_2, \cdots$)；对于每个 $n > 0$，n 元谓词类 (变元为 $\boldsymbol{X}_1^n, \boldsymbol{X}_2^n, \cdots$)；对于每个 $n > 0$，n 元函数类 (变元为 $\boldsymbol{F}_1^n, \boldsymbol{F}_2^n, \cdots$). 等号 (=) 仅在个体变元类的项之间使用. 我们在二阶语言中设定的谓词和函数参量仍然是多类语言中的参量，也将成为个体变元类的项. (对于一个函数参量 f，$f\vec{t}$ 是个体变元类的项. 谓词或函数类的项仅仅是这些类的变元.)

另外，我们现在开始使用两类新的参量. 对于每个 $n > 0$，存在一个 *属于* 谓词参量 ε_n，可以把它看作 n 元谓词类的一个项 (即，一个变元 $\boldsymbol{X}_m^n$) 和个体变元类的 n 个项. 所以，例如 [299]

$$\varepsilon_3 \boldsymbol{X}^3 v_2 v_1 v_8$$

是合式公式. 它的解释是，$\langle v_2, v_1, v_8 \rangle$ 表示的三元组在 $\boldsymbol{X}^3$ 表示的关系中. 这和二阶公式

$$\boldsymbol{X}^3 v_2 v_1 v_8,$$

的解释正好相同. 实际上，大家可以把这两个公式 *等同* 起来.

对于每个 $n > 0$，还存在着 *赋值* 函数参量 E_n，E_n 可以看作 n 元函数类中的项 (即，变元 $\boldsymbol{F}_m^n$) 和个体变元类的 n 个项. 表达式

$$E_n \boldsymbol{F}^n t_1 \cdots t_n,$$

就是个体变元类的项. 同样，大家可以把项 $E_n \boldsymbol{F}^n t_1 \cdots t_n$ 和前面的 $\boldsymbol{F}^n t_1 \cdots t_n$ 等同起来.

这样看来，存在一个互译 4.1 节的二阶语言和现在多类语言的自然方法. 有时候，我们坚持使用 ε_n 和 E_n，有时候又可以不用它们. 我们使用这些符号的目的是为了符合 4.3 节中语言的要求.

多类结构的每个类都有论域，并且 (和上一节一样) 给各类参量指派了合适的对象. 首先，我们要不失一般性地证明，可以用真正的属于关系来解释 ε_n，也可以用真正的赋值来解释 E_n.

定理 44A 设 $\mathfrak{A}$ 是上述多类语言的结构，并且满足 $\mathfrak{A}$ 的不同论域是不交的。那么存在一个把 $\mathfrak{A}$ 映射到 $\mathfrak{B}$ 上的同态 h，使得

(a) h 在个体论域上是一对一的，实际上是恒等的. (从这可以得到，对于每个公式 φ

$$\models_{\mathfrak{A}} \varphi[s] \quad \text{iff} \quad \models_{\mathfrak{B}} \varphi[h \circ s])$$

(b) $\mathfrak{B}$ 的 n 元谓词论域由个体论域上的 n 元关系组成，并且 $\langle R, a_1, \cdots, a_n \rangle$ 在 $\varepsilon_n^{\mathfrak{B}}$ 中当且仅当 $\langle a_1, \cdots, a_n \rangle \in R$.

(c) $\mathfrak{B}$ 的 n 元函数论域由个体论域上的 n 元函数组成，并且 $E_n^{\mathfrak{B}}(f, a_1, \cdots, a_n) = f(a_1, \cdots, a_n)$.

300

证明　由于 $\mathfrak{A}$ 的各类论域是不相交的，所以我们可以依次在各个类论域上定义 h. 在个体论域 $\mathfrak{A}$ 上，h 是恒等映射. 在 n 元谓词类的论域上，

$$h(Q) = \{\langle a_1, \cdots, a_n\rangle | \text{每个 } a_i \text{ 都在 } U \text{ 中并且 } \langle Q, a_1, \cdots, a_n\rangle \text{ 在 } \varepsilon_n^{\mathfrak{A}} \text{ 中}\}.$$

这样，

$$\langle a_1, \cdots, a_n\rangle \in h(Q) \quad \text{iff} \quad \langle Q, a_1, \cdots, a_n\rangle \text{在 } \varepsilon_n^{\mathfrak{A}} \text{ 中}. \tag{1}$$

类似地，在 n 元函数类的论域上，

$h(g)$ 是 U 上的 n 元函数，它在 $\langle a_1, \cdots, a_n\rangle$ 上的值为 $E_n^{\mathfrak{A}}(g, a_1, \cdots, a_n)$.

这样，

$$h(g)(a_1, \cdots, a_n) = E_n^{\mathfrak{A}}(g, a_1, \cdots, a_n). \tag{2}$$

对于 $\varepsilon_n^{\mathfrak{B}}$，我们取简单的属于关系，

$$\langle R, a_1, \cdots, a_n\rangle \text{在 } \varepsilon_n^{\mathfrak{B}} \text{ 中} \quad \text{iff} \quad \langle a_1, \cdots, a_n\rangle \in R. \tag{3}$$

对于 $E_n^{\mathfrak{B}}$，我们取赋值函数，

$$E_n^{\mathfrak{B}}(f, a_1, \cdots, a_n) = f(a_1, \cdots, a_n). \tag{4}$$

$\mathfrak{B}$ 的其他参量 (从二阶语言中继承的) 和 $\mathfrak{A}$ 相同.

显然，h 是 $\mathfrak{A}$ 到 $\mathfrak{B}$ 上的同态. 从 (1) 和 (3) 可以看出 h 保持了 ε_n，其中在 (3) 中取 $R = h(Q)$. 类似地，从 (2) 和 (4) 可以得到 h 保持了 E_n.

最后，我们要验证 (a) 中的附加说明. 我们知道只有个体变元类中有等号，h 在其上是一对一的. 利用这个事实，多类语言也有 2.2 节中同态定理的对应结果. 因此我们就能得到附加说明. ■

根据上面的定理，我们可以把注意力集中到结构 $\mathfrak{B}$ 上，其中的 ε_n 和 E_n 被定理的 (b) 和 (c) 确定. 但由于 $\varepsilon_n^{\mathfrak{B}}$ 和 $E_n^{\mathfrak{B}}$ 是由 $\mathfrak{B}$ 的剩余部分决定的，所以我们实际上并不需要它们. 当把它们去掉时，就得到了二阶语法的广义预备结构.

4.4.2　二阶语言的广义结构

这些结构给出了本节开始时提到的另一种语义.

定义　二阶语言的*广义预备结构* $\mathfrak{A}$ 包括一个 (原始意义上的) 结构和下列附加集合：

(a) 对于每个 $n > 0$，*n 元关系论域*是 $|\mathfrak{A}|$ 上的 n 元关系的集合.

301

(b) 对于每个 $n > 0$，*n 元函数论域*是从 $|\mathfrak{A}|^n$ 到 $|\mathfrak{A}|$ 内的函数集合.

另外，如果所有概括句子在 $\mathfrak{A}$ 中的取值都为真，则称 $\mathfrak{A}$ 为*广义结构*.

我们解释一下定义中的最后一句话. 首先概括句子是指对概括公式进行概化所得到的句子 (见 4.1 节中的例 3.). 因此它可以是

$$\exists \boldsymbol{X}^n \forall v_1 \cdots \forall v_n (\boldsymbol{X}^n v_1 \cdots v_n \leftrightarrow \varphi),$$

其中 $\boldsymbol{X}^n$ 不在 φ 中自由出现，或

$$\begin{aligned}&\forall v_1\cdots\forall v_n\exists!v_{n+1}\psi\rightarrow\\&\quad\exists\boldsymbol{F}^n\forall v_1\cdots\forall v_{n+1}(\boldsymbol{F}^n v_1\cdots v_n=v_{n+1}\leftrightarrow\psi),\end{aligned}$$

其中 $\boldsymbol{F}^n$ 不在 ψ 中自由出现.（这里的 φ 和 ψ 可以含有个体变元，谓词变元和函数变元.）

其次我们要说明概括句子在 $\mathfrak{A}$ 中的取值都为真的含义。假设 $\mathfrak{A}$ 是一个广义预备结构，那么句子 σ 在 $\mathfrak{A}$ 中的取值为真，当且仅当 σ 译成的多类句子 (添加 ε_n 和 E_n) 在 $\mathfrak{A}$ 中的取值为真，ε_n 用属于关系来解释，E_n 用赋值关系来解释。

更一般地，令 φ 为二阶公式，s 为一个函数，它给每个个体变元指派 $|\mathfrak{A}|$ 中的一个元素，给每个谓词变元指派 $\mathfrak{A}$ 的论域中的关系，给每个函数变元指派一个 $\mathfrak{A}$ 的论域中的函数. 我们称 $\mathfrak{A}$ 由 s 满足 φ(记作 $\models^G_{\mathfrak{A}}\varphi[s]$)，当且仅当 φ 的多类形式在结构 $\mathfrak{A}$ 中由 s 满足，其中 ε_n 用属于关系来解释，E_n 用赋值关系来解释.

下面的句子本质上是从满足这个概念中得到的，可以把它们和 214 页的 5 和 6 进行比较.

$\models^G_{\mathfrak{A}}\forall\boldsymbol{X}^n\varphi[s]$ iff 对于 $\mathfrak{A}$ 的 n 元关系论域中的任意一个 R，$\models^G_{\mathfrak{A}}\varphi[s(\boldsymbol{X}^n|R)]$.

$\models^G_{\mathfrak{A}}\forall\boldsymbol{F}^n\varphi[s]$ iff 对于 $\mathfrak{A}$ 的 n 元函数论域中的任意一个 f，$\models^G_{\mathfrak{A}}\varphi[s(\boldsymbol{F}^n|f)]$.

这就是本节开头提到的另一种方法. 它本质上是把二阶语言看作一阶多类语言. 由于这种方法本质上是一阶的，因此洛文海 - 斯科伦定理、紧致性定理和可枚举性定理都成立.

洛文海 – 斯科伦定理 如果可数二阶语言中的句子集 Σ 有广义模型，那么它有可数广义模型。

302

这里的可数广义模型是指模型的每个论域都是可数的 (或者等价地，所有论域的并是可数的).

证明 令 Γ 是概括句子的集合，那么当把 $\Sigma\cup\Gamma$ 看作多类句子的集合时，根据前一节的洛文海 - 斯科伦定理，它有可数的多类模型. 再根据定理 44A，这个模型的同态像就是满足 $\Sigma\cup\Gamma$ 的广义预备结构，因此是 Σ 的广义模型. ■

紧致性定理 如果二阶命题集 Σ 的每个有限子集都有广义模型，那么 Σ 有广义模型.

证明 证明和上面一个定理的证明相似. $\Sigma\cup\Gamma$ 的每个有限子集都有多类模型，因此我们可以使用前一节的紧致性定理。 ■

可枚举性定理 假设语言是可数递归的，那么在每个广义结构中取值为真的二阶命题的哥德尔数集是递归可枚举的.

证明 句子 σ 在每个广义结构中的取值为真，当且仅当它是 Γ 的多类推论，并且 $\sharp\Gamma$ 是递归的。 ■

上面的两个定理保证，存在可接受的演绎算法使得 τ 可从 Σ 推出当且仅当 τ 在 Σ 的每个广义模型中的取值都是真的 (见 2.4 节开头的注释). 但是既然存在这样一个完全的演绎算法，我们就不必研究算法的细节了.

下面我们从二阶语义的角度比较一下这两种方法：4.1 节中的形式 (我们现在可以称其为 *绝对* 二阶逻辑) 是混合产物，其中参量的含义要根据结构来给出，但作为子集的概念是确定的，不会改变. 本节中的 (广义二阶逻辑) 形式避免涉及子集的明确定义，从而可简化为一阶逻辑. 从这个关系看，这有点像公理集合论，在公理集合论中我们用到集合、集合的集合等，但其理论仍然是一阶理论.

通过扩充结构的类，广义二阶逻辑中逻辑蕴涵成立的情况减少了. 也就是说，如果 Σ 的每个广义模型都是 σ 的广义模型，那么在绝对二阶逻辑中 $\Sigma \vDash \sigma$，但反之是不成立的. 例如，取 $\Sigma = \varnothing$：在所有广义模型中取值为真的句子集是绝对二阶逻辑中恒真句子的非算术集的递归可枚举子集.

303

4.4.3 解析模型

我们可以通过研究一个非常有趣的例子来说明本节的思想，这个例子是二阶数论的广义模型. 现在考虑含有参量 $\mathbf{0}, \mathbf{S}, <, \cdot$ 和 $\mathbf{E}$ 的数论二阶语言，公理集为 A_E 中添加皮亚诺归纳假设 (例 2, 4.1 节) 后的集合 A_E^2. 从 4.1 节的习题 1 中我们能够得出 A_E^2 同构于 $\mathfrak{N}$.

但上面提出的公理集的广义模型是什么样的？我们说它们和 $\mathfrak{N}$ 在两个方面有区别. 我们可以用紧致性定理来构造含有无限个元素 (非标准) 的广义模型 (即，模型 $\mathfrak{A}$ 在序 $<^{\mathfrak{A}}$ 下总含有比 $\mathbf{S}^n\mathbf{0}$ 大的元素). 我们也能找到 (非绝对的) 广义模型，它中的集合论域 (一元关系论域) 要比个体论域的幂集小. 确实，任意可数的广义模型一定属于这个类.

习惯上，逻辑学家们把二阶数论称为 *解析*. 取这个名称是因为有可能通过这个方法用自然数来定义实数. 在二阶数论中，可以对自然数的集合使用量词，我们可以把它看作对实数使用量词. 无论这个名称叫什么，它的用途是确定的. 通过 *解析模型*，我们能够得到上述公理 A_E^2 的广义模型.

定义 *解析ω模型* 为个体论域是 $\mathbb{N}$，用 $\mathbf{0}$ 和 $\mathbf{S}$ 来解释真正的 $\mathbf{0}$ 和 $\mathbf{S}$ 的解析模型. (进而，$<, \cdot$ 和 $\boldsymbol{E}$ 的解释也是标准的.) 我们研究 ω 模型是为了了解 $\mathbb{N}$ 的幂集. 但是我们 (自认为) 已经清楚地了解了 $\mathbb{N}$，但对它的幂集 $\mathcal{P}\mathbb{N}$ 却知之甚少. 例如，我们不知道它的基数是 $\aleph_1$ 还是 $\aleph_2$，还是更大. 因此，我们有必要了解一下与 $(\mathcal{P}\mathbb{N})$ 有关的结构.

在解析的 ω 模型中，有一种模型称为 *绝对模型*，它的 n 元关系论域包含了 $\mathbb{N}$ 上所有的 n 元关系 (它的函数论域包含所有可能的函数). 一个一阶命题在任意一个解析 ω 模型中取值都为真当且仅当它在 $\mathfrak{N}$ 中取值为真. 但 ω 模型可能和二阶命题上的绝对模型不同.

在下一个定理中我们将断言，解析 ω 模型完全由它的集合论域 (即，它的一元关系论域) 所决定.

304

定理 44B 如果 $\mathfrak{A}$ 和 $\mathfrak{B}$ 是具有相同一元关系论域的解析 ω 模型，那么 $\mathfrak{A} = \mathfrak{B}$.

证明 假设 R 属于 $\mathfrak{A}$ 的三元关系论域，令 $\langle R\rangle$ 是 R 到一元关系的"压缩"：

$$\langle R\rangle = \{\langle a,b,c\rangle | \langle a,b,c\rangle \in R\}.$$

由于编码序列函数是递归的，因此可以在数论中由一阶公式 φ 来定义. 根据概括句子

$$\forall \boldsymbol{X}^3 \exists \boldsymbol{X}^1 \forall u[\boldsymbol{X}^1 u \leftrightarrow \exists v_1 \exists v_2 \exists v_3(\varphi(v_1,v_2,v_3,u) \wedge \boldsymbol{X}^3 v_1 v_2 v_3)].$$

知，$\langle R\rangle$ 在 $\mathfrak{A}$ 的集合论域中.

这样，通过同样的讨论可知，$\langle R\rangle$ 在 $\mathfrak{B}$ 的集合论域中. 根据概括句子

$$\forall \boldsymbol{X}^1 \exists \boldsymbol{X}^3 \forall v_1 \forall v_2 \forall v_3 [\boldsymbol{X}^3 v_1 v_2 v_3 \leftrightarrow \exists u(\varphi(v_1, v_2, v_3, u) \wedge \boldsymbol{X}^1 u)].$$

知，R 在 $\mathfrak{B}$ 的三元关系论域中.

类似的讨论可以运用到函数论域上. ■

因此，我们可以通过集合论域 (包含在 $\mathcal{P}\mathbb{N}$ 中) 来确定解析 ω 模型. 但并不是 $\mathcal{P}\mathbb{N}$ 的每个子集都是一个解析 ω 模型，而只有那些由概括句子满足的子集才是.

ω 模型的例子 我们只需指出集合论域就行了.

(1) $\mathcal{P}\mathbb{N}$ 是绝对模型.

(2) 设 $(A;\in_A)$ 是集合论常用公理的模型，并且满足 (i) 关系 $\in_A$ 是论域 A 上的真正属于关系 $\{\langle a,b\rangle | a\in A, b\in A$ 并且 $a\in b\}$, (ii)A 是传递的，即，如果 $a\in b\in A$，则 $a\in A$. 那么所有属于 A 的 $\mathbb{N}$ 的子集的并是解析的 ω 模型.

(3) 对于类 $\mathcal{A}\subseteq\mathcal{P}\mathbb{N}$，定义 $\mathbb{D}\mathcal{A}$ 为所有这样的 $B\subseteq\mathbb{N}$ 的类，其中 B 是由二阶数论语言中的一个公式在 ω 预备结构的集合论域 $\mathcal{A}$ 中定义的集合，这个类随着 $\mathcal{A}$ 中的每个集合参数的变化而变化. 由序数上的无限归纳法，我们有：

$$\begin{aligned} \mathcal{A}_0 &= \varnothing, \\ \mathcal{A}_{\alpha+1} &= \mathbb{D}\mathcal{A}_\alpha, \\ \mathcal{A}_\lambda &= \cup_{\alpha<\lambda}\mathcal{A}_\alpha \quad \text{对于极限}\lambda. \end{aligned}$$

从基数的角度考虑，这个过程在某个使得 $\mathcal{A}_{\beta+1}=\mathcal{A}_\beta$的序数 β 处停止. 令 β_0 为这种 β 的最小的一个，由洛文海 - 斯科伦定理，可以证明 β_0 是可数序数. $\mathcal{A}_{\beta_0}$ 等于 $\bigcup_\alpha \mathcal{A}_\alpha$(关于所有序数 α 的并)，称为*分歧解析集*. 它是一个解析 ω 模型，从事实 $\mathbb{D}\mathcal{A}_{\beta_0}\not\subseteq\mathcal{A}_{\beta_0}$ 可以得到概括句子的赋值为真. [305] [306]



推荐读物

Jon Barwise (editor). *Handbook of Mathematical Logic.* North-Holland Publishing Company, Amsterdam, 1978.

这本“手册”收录了 31 篇由逻辑学专家撰写的介绍文章，内容涉及模型论、集合论、递归论和证明论.

Jon Barwise and John Etchemendy. *The Language of First-order Logic.* Center for the Study of Language and Information, Stanford, 1992.

这本介绍性的教科书附带有一张 *Tarski' s world* 软件包的光盘. 软件包的开发者还设计了 *Turing's world* 和 *Hyperproof* 软件包.

J.L.Bell and M.Machover. *A Course in Mathematical Logic.* North-Holland Publishing Company, Amsterdam, 1977.

George Boolos and Richard Jeffrey. *Computability and Logic.* Cambridge University Press, Cambridge, 1974 (1989 年第 3 版).

这本书在新版本中给出了一些问题的实际解决方法，适合各类不同的读者.

C.C.Chang and H.J.Keisler. *Model Theory.* North-Holland Publishing Company, Amsterdam, 1973(1990 年第 3 版).

这是模型论的经典教材.

Herbert B. Enderton. *Elements of Set Theory.* Academic Press, New York, 1977.

这是本书作者自己撰写的也是最喜欢的有关集合论的书.

Wilfrid Hodges. *A Shorter Model Theory.* Cambridge University Press, Cambridge, 1997.

这是该作者写的 *Model Theory* 一书的简版，*Model Theory* 出版于 1993 年.

Hartley Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill Book Company, New York, 1967.

这本书仍是这个领域的经典之作.

Joseph R. Shoenfield. *Mathematical Logic.* Association for Symbolic Logic and A K Peters, Natick, Massachusetts, 2000.

曾由 Addison-Wesley 在 1967 年发行，这本书简洁紧凑，适合研究生水平的读者阅读.

Jean van Heijenoort (editor). *From Frege to Gödel : A Source Book in Mathematical Logic, 1879-1931.* Harvard University Press, Cambridge, Massachusetts, 1967.

这本书包含了逻辑学中的 46 篇奠基性的论文，英文版附有注释.

附录 B

符号列表

数字为符号首次出现的页码 (对应页边栏的页码).

索　引

索引中的页码为英文原书的页码，与书中边栏的页码一致。

K

L

M

T

U

V

W

Z